ЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

КАФЕДРА СИСТЕМНОГО ПРОГРАМУВАННЯ І
СПЕЦІАЛІЗОВАНИХ КОМП'ЮТЕРНИХ СИСТЕМ

«На правах рукопису»
УДК 004.9

«До захисту допущено»
Завідувач кафедри СПСКС

Віталій РОМАНКЕВИЧ

_____2020 р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 123 Комп'ютерна інженерія
(Комп'ютерні системи та компоненти)

на тему: Спосіб виявлення основних ключових фрагментів в неструктурованих текстах

Виконала: студентка II курсу, групи _КВ-91-мп
Мандрік Марія Владиславівна

Науковий керівник Орлова Марія Миколаївна, доцент кафедри СПСКС
к.т.н., доцент

Консультант з нормоконтролю доцент, с.н.с.,к.т.н. Юлія БОЯРІНОВА

Рецензент доцент, к.т.н., доцент Марія ОРЛОВА

Засвідчую, що у цій магістерській дисертації немає
запозичень з праць інших авторів без відповідних посилань.

Студент _____

Київ – 2020 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра системного програмування і спеціалізованих комп'ютерних систем

Рівень вищої освіти – другий (магістерський)
за освітньо-професійною програмою
Спеціальність 123 Комп'ютерна інженерія
Комп'ютерні системи та компоненти

ЗАТВЕРДЖУЮ
Завідувач кафедри СПСКС

_____Віталій РОМАНКЕВИЧ

«_01__»_____12_____2019р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Мандрік Марії Владиславівні

1. Тема дисертації Спосіб виявлення основних ключових фрагментів в неструктурованих текстах,
науковий керівник дисертації Орлова Марія Миколаївна, доцент кафедри СПСКС к.т.н. доцент,

затверджені наказом по університету від «12» листопада 2020 р. № 3298-С

2. Термін подання студентом дисертації 10 грудня 2020 р.

3. Об'єкт дослідження способи виділення ключових фраз з неструктурованих текстів

4. Предмет дослідження є розробка способу виділення ключових фактів на основі алгоритму TF-IDF та алгоритм пошуку ключових слів, заснований на частоті фактів, що виділяють інформацію з тексту а основі онтології.

5. Перелік завдань, які потрібно розробити: аналіз існуючих методів виділення кочових фрагментів, виділення основних рис з існуючих алгоритмів, розробка нового алгоритму виділення ключових фрагментів, програмна реалізація розробленого алгоритму

6. Перелік ілюстративного матеріалу: презентаці.

7. Перелік публікацій За тематикою проведених досліджень опубліковано 2 наукові праці, а саме тези доповідей на конференціях.

8. Дата видачі завдання 5 вересня 2019 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Формування мети та цілі роботи	01.11.2019	
2	Дослідження теоретичного матеріалу	01.03.2020	
3	Дослідження існуючих алгоритмів	01.05.2020	
4	Розробка нового алгоритму	01.07.2020	
5	Реалізація нового алгоритму	01.09.2020	
6	Проведення аналізу роботи алгоритму	01.10.2020	
7	Написання дисертації	20.11.2020	
8	Попередній розгляд магістерської дисертації на кафедрі	26.11.2020	

Студент _____ Марія МАНДРІК

Науковий керівник дисертації _____ Марія ОРЛОВА

РЕФЕРАТ

Актуальність теми. Для надання інформації про товари і послуги, яка відповідає пошуковим запитам користувачів і має високий ступінь пертінентності, багато Інтернет-порталів впроваджують відповідні рекомендаційні сервіси. Найпростішим і найбільш поширеним різновидом таких рекомендаційних сервісів є системи фільтрації, засновані на фіксованому наборі параметрів-фільтрів, організованих у форму введення на сайті.

Для здійснення фільтрації туристичних продуктів за фіксованим набором параметрів, як правило, використовується база даних, що містить набір характеристик продукту. При цьому впровадження такої бази знань в сервіс накладає зобов'язання з підтримки її актуальності, а також регулярної перевірці та коригуванні інформації, що міститься в ній. Як правило, здійснення цих завдань вимагає залучення контент-менеджерів для виконання значної кількості «ручної» праці. Як відгуки туристів, так і новинні замітки є неструктурованими текстами, представленими природною мовою. Для ефективного аналізу вмісту подібних текстів використовуються так звані «факти» - пари слів виду «параметр» (головне, слово, що визначає зміст) + «характеристика» (залежне слово, визначення), - які можна отримати з тексту за допомогою різних інструментів синтаксичного аналізу.

Мета роботи полягає у підвищенні ефективності пошуку ключових фраз у неструктурованих текстах, що представлені природною мовою, зокрема українською, за рахунок автоматизацій деяких процесів виділення ключових фраз та залучення словників фраз з певної тематики, що збільшує швидкість обробки текстів без залучення людини.

Для досягнення поставленої мети в роботі вирішуються наступні задачі.

1. Дослідження існуючих елементів для роботи з текстами написаними природною мовою.

2. Розробка методу вилучення ключових фрагментів з текстів природною мовою.

3. Реалізація розробленого алгоритму для статей та відгуків про готелі.

Об'єктом дослідження є способи виділення ключових фраз з неструктурованих текстів.

Предметом дослідження є способи виділення ключових фактів на основі алгоритму TF-IDF та алгоритму пошуку ключових слів, який базується на частоті фактів, що виділяють інформацію з тексту а основі онтології.

Методи дослідження. В роботі використовуються методи оптимізації, методи системного аналізу, а також методів моделювання.

Наукова новизна одержаних результатів полягає в тому, що підвищена ефективність виявлення ключових фактів з неструктурованих текстів, розроблений на його основі алгоритм, який нівелює недоліки вже існуючих, а саме:

1. Виявляє кочові фрази за заданою тематикикою.
2. Оцінює достовірність вилучених фактів.

Практична цінність одержаних результатів зводиться до виділення ключових слів з статей та відгуків про готелі українською мовою, що спрощує подальшу роботу з ними, тобто побудову коротшого та більш чіткого опису готелів і спрощує пошук потрібних для клієнтів характеристик готелів.

Апробація роботи. Основні положення і результати роботи були представлені та обговорювались на:

- XIII науковій конференції молодих вчених «Прикладна математика та комп'ютинг» ПМК-2020;
- VI міжнародна науково-технічна Internet-конференція.

Публікації. За тематикою проведених досліджень опубліковано 2 наукові праці, а саме тези доповідей на 2-х конференціях.

Структура та обсяг роботи.

Магістерська дисертація складається з вступу, трьох розділів, висновків та додатків.

У вступі надано загальну характеристику програмного коду, проблематику виділення ключових фраз з неструктурованих текстів, сформульовано мету дослідження, показано практичну цінність роботи.

У першому розділі надано детальне обґрунтування актуальності напрямку досліджень, виконано оцінку поточного стану в даній сфері, представлено теоретичний огляд виділення ключових фраз з неструктурованих текстів.

У другому розділі розроблено та описано спосіб виділення фраз з неструктурованих тестів українською мовою.

У третьому розділі проведено апробацію.

У висновках проаналізовано отримані результати роботи.

Ключові слова: ключові фрази, алгоритм TF-IDF, частота фактів, неструктуровані тексти, природна мова.

ABSTRACT

Actuality of theme. There is a high degree of consistency in providing information about products and services that is responsible for users' search queries, and many Internet portals implement responses to service recommendations. The simplest and most common type of such recommendation services is a filtering system, fixed on a fixed set of filter parameters, organized in the form of input to the Internet.

To perform the filtering of tourist products for fixed sets of parameters, as a rule, a database containing a set of product characteristics is used. In this way, the introduction of the knowledge base in the service staff is associated with maintaining its relevance, as well as regular viewers and information on the use contained in it. As a rule, the use of these tasks requires the involvement of content managers to perform a significant amount of "manual" work. Both tourist reviews and replacement news are unstructured texts presented in natural language. For effective analysis of the content of such texts are used so-called "facts" - pairs of words in the form of "parameter" (main, the word that defines the content) + "characteristic" (dependent word, definition) - which can be obtained from the text using various tools parsing.

The purpose of the work is to increase the efficiency of searching for key phrases in structured texts, presented in natural language, Ukrainian, to automate certain processes of selection of key phrases and to involve dictionaries on certain topics, which increases the speed of word processing without human intervention.

To achieve this goal, the following tasks are solved in the work.

1. Research of existing elements for work with texts written in natural language.

2. Development of a method of extracting key fragments from the text of a natural word.

3. Implementation of the developed algorithm for the state and change of residence.

The object of the study are ways to extract key phrases from unstructured texts.

The subject of the research is the methods of highlighting key facts based on the TF-IDF algorithm and the keyword search algorithm, which are based on the frequency of facts that extract information from the text and the basis of the ontology.

Research methods. The paper uses optimization methods, methods of systems analysis, graph theory, as well as modeling methods.

The scientific novelty of the obtained results is that the efficiency of detection of key facts from unstructured texts is increased, the algorithm developed on its basis which eliminates shortcomings of already existing, namely:

1. Detects nomadic phrases on a given topic.

2. Evaluates the authenticity of the removed facts.

The practical novelty of search results search results to the selection of keywords from articles and descriptions of living in the Ukrainian language, which facilitates further work with them, creating a shorter and clearer description of hotels and search queries for customers.

Approbation of work. The main provisions and results of the work were presented and discussed at the XIII Scientific Conference of Young Scientists "Applied Mathematics and Computing" PMK-2020.

Publications. 2 scientific papers were published on the subject of the conducted researches, namely these reports at 2 conferences.

Structure and scope of work.

The master's dissertation is created from the introduction, three sections, the conclusion and appendices.

In the introduction the general characteristic of the program code, problems of allocation of key phrases from unstructured texts is given, the purpose of research is formed, practical value of work is shown.

The first section provides a detailed report on the relevance of research results, evaluates the current situation in this area, presents a theoretical overview of the

selection of key phrases from unstructured texts.

Another section develops and describes a method of extracting phrases from unstructured tests in the Ukrainian language.

In the third section the approbation is carried out.

The results of the work are analyzed in the conclusions.

Keywords: key phrases, TF-IDF algorithm, frequency of facts, unstructured texts, natural language.

Зміст:

Список термінів, скорочень та позначень

Автокомпліт — (англ. Autocomplete) - функція в програмах, які передбачають інтерактивний введення тексту (редактори, оболонки командного рядка, браузери і т. Д.) Щодо доповнення тексту по введеної його частини.

Алгоритм Гірвана-Ньюмана — ієрархічний метод, який використовується для виявлення структур співтовариств в складних системах. Розроблений американським математиком Мішель Гірван і британським фізиком Марком Ньюменом.

Атрибут — іменований елемент певного типу в класі, який використовується для представлення інформації про модельованої сутності.

БД – база даних.

Верифікація — доказ того, що вірогідній факт або тверджень є істинним.

Вікіпедія — загальнодоступна багатомовна універсальна інтернет-енциклопедія з вільним контентом.

Жаргонізми — жаргонні слова або виразу.

Парсер — (англ. Parcer) це програма, сервіс або скрипт, який збирає дані з зазначених веб-ресурсів, аналізує їх і видає в потрібному форматі.

ПЗ – програмне забезпечення.

Скрипт — це окремі послідовності дій, створені для автоматичного виконання завдання.

Скріншот — зображення, отримане пристроєм і показує в точності те, що бачить користувач на екрані монітора або іншого візуального пристрої виведення.

Спам — масова розсилка кореспонденції рекламного характеру особам, які не виражав бажання її отримати, а також розсилка масових повідомлень.

Токен — об'єкт, що створює з лексеми в процесі лексичного аналізу.

Токенізація — це процес видылення з тексту на компонентів, зокрема на

токени.

Чат бот — це програма, яка імітує реальний розмова з користувачем. Чат-боти дозволяють спілкуватися за допомогою текстових або аудіо повідомлень на сайтах, в месенджерах, мобільних додатках або по телефону.

AI — (англ. artificial intelligence, AI) властивість інтелектуальних систем виконувати творчі функції, які традиційно вважаються прерогативою людини; наука і технологія створення інтелектуальних машин, особливо інтелектуальних комп'ютерних програм.

API — це сукупність засобів та правил, що вможливлюють взаємодію між окремими складниками програмного забезпечення або між програмним та апаратним забезпечення.

Azure — хмарна платформа компанії Microsoft.

GLR-парсинг Generalized left-to-right algorithm — в інформатиці розширений алгоритм LR-парсера, призначений для розбору по недетермінованим і неоднозначним граматика.

IEEE — група стандартнів.

IVR — система попередньо записаних голосових повідомлень, що виконує функцію маршрутизації дзвінків всередині call-центру з використанням інформації, що вводиться клієнтом на клавіатурі телефону за допомогою тонального набору.

LR-парсинг — синтаксичний аналізатор для вихідних кодів програм, написаних на деякій мові програмування, який читає вхідний потік зліва (Left) направо і виробляє найбільш праву (Right) продукцію контекстно-вільної граматики.

NLP — (англ. Natural Language Processing, NLP) загальний напрямок штучного інтелекту і математичної лінгвістики. Воно вивчає проблеми комп'ютерного аналізу і синтезу текстів на природних мовах.

Perl — високорівнева динамічний мова програмування загального призначення.

RDF — це модель даних, яка використовується для подання ресурсів т.зв. семантичної павутини (semantic web).

The Daily Telegraph - щоденна британська газета.

Wi-fi — технологія бездротового локальної мережі з пристроями на основі стандартів IEEE 802.11.

ВСТУП

З кожним роком мережа Інтернет все тісніще входить у повсякденне життя людей. За допомогою різноманітних засобів всесвітньої мережі люди мають змогу спілкуватися, робити покупки, дивитися фото та відео матеріал, діставати нову інформацію не виходячи з дому.

За даними статистики, опублікованими Світовим банком, за останні десять років частка жителів України, що мають доступ до мережі Інтернет, перевищила 63% [1]. У той же самий час, згідно з дослідженнями, проведеними аналітичним центром Pew Research Center в 2014 році і присвяченим використання інформаційних технологій і, зокрема, Інтернету в країнах, що розвиваються, одним з основних напрямків використання Інтернету є пошук інформації про різні продукти і здійснення онлайн-покупок [2].

Для надання інформації про товари і послуги, яка відповідає пошуковим запитам користувачів і має високий ступінь пертінентності, багато Інтернет-портали впроваджують відповідні рекомендаційні сервіси. Найпростішим і найбільш поширеним різновидом таких рекомендаційних сервісів є системи фільтрації, засновані на фіксованому наборі параметрів-фільтрів, організованих в форму введення на сайті. В цьому випадку процес побудови рекомендації складається з обробки заданих користувачем параметрів і подальшої побудови відповідного рейтингу. Одним з найпопулярніших сервісів в російськомовному Інтернеті, що використовують описану рекомендаційну модель, є «Booking».

У зв'язку з популярністю туризму як в Україні, так і в усьому світі, а також проникненням телекомунікаційних технологій в туристичну галузь готельні послуги, тури і екскурсії поповнили перелік продуктів, інформацію про які люди вважають за краще отримувати з мережі Інтернет. Цим зумовлена поява і стрімкий розвиток таких міжнародних Інтернет-сервісів, як Booking.com і TripAdvisor. У українському сегменті можна виділити TopHotels. Всі зазначені

сервіси використовують фільтри при створенні своїх рекомендацій. Наприклад, готелі можна вибирати за такими характеристиками як близькість до пляжу, наявність бездротового або дротового Інтернету, тип харчування, місце розташування та інші. В даній роботі в якості прикладу туристичних продуктів будуть розглядатися такі види готельних будинків, як готелі, хостели, напівпансіон і апартаменти. Для полегшення сприйняття всі перераховані вище різновиду будуть позначатися словом «готель».

Для здійснення фільтрації туристичних продуктів по фіксованому набору параметрів, як правило, використовується база даних, що містить набір характеристик продукту. При цьому впровадження такої бази знань в сервіс накладає зобов'язання з підтримки її актуальності, а також регулярної перевірки та коригуванні того, міститься в ній інформації. Як правило, здійснення цих завдань вимагає залучення контент-менеджерів для виконання значної кількості «ручної» праці. Таким чином перспективним виглядає внесення змін в існуючу базу даних, засноване на автоматичній обробці постійно оновлюються інформаційних ресурсів. У сфері туризму такими ресурсами є відгуки мандрівників на спеціалізованих сайтах, а також статті в новинних Інтернет-виданнях.

Як відгуки туристів, так і новинні замітки є неструктурованими текстами на природній мові. Для ефективного аналізу вмісту подібних текстів використовуються так звані «Факти» - біграми виду «параметр» (головне, визначається слово) + «Характеристика» (залежне слово, визначення), - які можна отримати з тексту за допомогою різних інструментів синтаксичного аналізу. Одним з широко використовуваних інструментів такого роду є Томіта-парсер, розроблений на основі алгоритму GLR-парсинга, описаного японським ученим Масару Томіта [3]. Однак, число фактів, витягнутих з тексту з використанням подібних «синтаксичних» інструментів часто надмірне:

наприклад, тому що працюючий на основі граматик і словників Томіта-парсер витягує зі статті довжиною в 5000 символів близько сотні словосполучень, в той час як її зміст може бути засноване не більше ніж на десяти основних фактах - так званих «ключових фактах». Автоматизація вилучення саме скороченої множини фактів в рамках деякої заданої тематики вимагає особливого підходу при обробці текстів на природній мові.

1. ОГЛЯД ІСНУЮЧИХ РІШЕНЬ ТА ОБГРУНТУВАННЯ ТЕМИ МАГІСТЕРСЬКОЇ ДИСЕРТАЦІЇ

1.1 Вилучення синтаксичних фактів з текстів природною мовою

Вилучення ключових фраз з тексту має широке застосування в різних областях. Ключові фрази - це набір слів або словосполучень, які відображають основну тему документа і є найкращою характеристикою його вмісту. Ключові фрази як правило необхідні для побудови пошукових індексів і класифікаторів різних текстових документів.

У сучасному світі велика кількість різної текстової інформації вимагає класифікації зберігається в електронному вигляді. Кількість таких документів постійно збільшуються. Однією з важливих задач є класифікація текстових документів для можливості їх швидкого і точного пошуку за заданими критеріями.

На сьогоднішній день існують інструменти для виділення з текстів природною мовою різних синтаксично коректних словосполучень, відповідних граматиці цієї мови. Нижче описані конкретні інструменти для української та англійської мов, що використовуються в промислових сервісах, а також в дослідженнях найбільш часто.

Завдання автоматичного вилучення ключових фраз з тексту складається з декількох етапів:

1. Попередня обробка тексту.
2. Відбір кандидатів ключових фраз.
3. Розрахунок ознак для кожного кандидата.
4. Відбір ключових фраз з числа кандидатів.

У процесі попереднього відпрацьовування з тексту проводиться видалення неінформативних частин (малюнки, таблиці тощо).

Кандидати ключових фраз відбираються у вигляді N-грам, що не розділених знаками пунктуації (крім дефіса і лапок) і стоп-словами. Де N-грами - це термін з комп'ютерної лінгвістики, що означає послідовність з N елементів тексту, наприклад слово або їх послідовність. Стоп-слова - слова не несуть ніякого смислового навантаження, прийменники, сполучники, вигуки, які часто зустрічаються в будь-якому документі.

Для кожного з кандидатів у список ключових фраз розраховуються ознаки, які дозволяють судити про важливість даного кандидата в даному тексті. Набір кандидатів у ключові фрази впорядковується за значеннями ознак, наприклад відповідно до їх частотністю і вагами інформативності, розрахованими по одній з методик.

Ключові слова — це одно- і багатокомпонентні лексичні групи, що відображають зміст документа. Автоматичне виділення ключових слів є необхідним етапом обробки тексту в таких важливих додатках, як системи автоматичного інформаційного пошуку, анотування, реферування і т. д. Однак, незважаючи на досить велику кількість досліджень, автоматичне вилучення ключових слів являє собою досить велику проблему, яка до сих пір не вирішена. Проблематичним є автоматичне вилучення багатокомпонентних ключових слів, особливо, якщо робиться спроба автоматично отримати певні типи лексичних груп, наприклад, іменні групи. Для всіх методик алгоритм верхнього рівня виділення ключових слів універсальний і включає етапи:

- формування 23 «кандидатів» в ключові слова і
- фільтрації цієї множини для отримання результуючого списку ключових слів.

Виділяють 3 групи методів виділення ключових слів: статистичні, лексичні та гібридні.

Існує метод автоматичного вилучення ключових термінів з текстових

документів, заснований на мірі семантичної близькості термінів і обчислюєьбся за допомогою використанням бази даних, побудові семантичного графа, виборі тематичних термінів за допомогою алгоритму Гірвана-Ньюмана. Одним з переваг цього методу, на думку його авторів, є відсутність необхідності в попередньому навчанні, так як алгоритм працює безпосередньо з базою даних. Експериментальні оцінки ефективності методу показують високу точність і повноту вилучення з тексту ключових термінів. Лінгвістичні методи, засновані на застосуванні використання значень слів, словників, онтологій, енциклопедій, в тому числі, Вікіпедії, деякі дослідники пропонують виділити в окремий метод метод на основі баз даних і значень слів.

Обробка природної мови зараз не використовуються хіба що в зовсім консервативних галузях. У більшості технологічних рішень розпізнавання і обробка «людських» мов давно впроваджена: саме тому звичайний IVR з жорстко заданими опціями відповідей поступово відходить у минуле, чатбот починають все більш адекватно спілкуватися без участі живого оператора, фільтри в пошті працюють відмінно тощо. Розглянемо як відбувається розпізнавання записаної мови, тобто тексту. Основним питанням є, що лежить в основі сучасних технік розпізнавання і обробки. На це добре відповідає наш сьогоднішній адаптований переклад тобто основні принципи технології NLP.

Natural Language Processing (далі - NLP) – технологія обробки природної мови - підрозділ інформатики і AI, присвячений тому, як комп'ютери аналізують природні (людські) мови, тобто ті,які люди використовують у повсякденному житті. NLP дозволяє застосовувати алгоритми машинного навчання для тексту й мови. Приклад застосування NLP в Azure наведено нижче на рисунку 1.

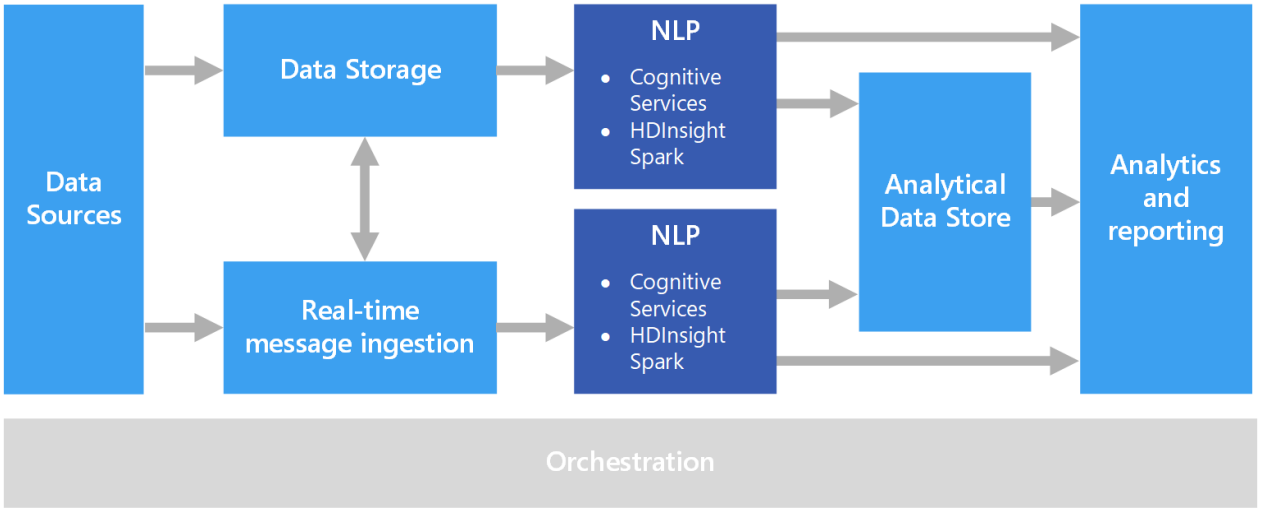


Рисунок 1 - Приклад застосування NLP

Наприклад, NLP часто використовується, щоб створювати системи на кшталт розпізнавання мови, узагальнення документів, машинного перекладу, виявлення спаму, розпізнавання іменованих сутностей, відповідей на питання, автокомпліта, інтелектуального введення тексту і т.д.

Сьогодні у багатьох з нас є смартфони та інші прилади з розпізнаванням мови - в них використовується NLP для того, щоб розуміти нашу мову. Також багато людей використовують ноутбуки з вбудованим в ОС розпізнаванням мови.

Існує декілька підходів до вирішення цієї проблеми, розглянемо кожен із них детальніше у наступних підпунктах.

1.1.1 Токенізація за словами

Токенізація (іноді - сегментація) за словами - це процес поділу пропозицій на слова-компоненти. В англійській і багатьох інших мовах, що використовують ту чи іншу версію латинського алфавіту, пробіл – роздільник слів, який використовується найчастіше.

Проте, можуть виникнути проблеми, якщо ми будемо використовувати тільки пробіл - в англійському складові іменники пишуться по-різному і іноді через пробіл. І тут знову нам допомагають бібліотеки.

1.1.2 Токенізація за пропозиціями

Токенізація (іноді - сегментація) за пропозиціями - це процес поділу писемної мови на пропозиції-компоненти. Ідея виглядає досить просто. В англійській і деяких інших мовах є можливість виокремлювати пропозицію кожен раз, коли знаходимо певний знак пунктуації - точку.

Але навіть в англійській мові ця задача нетривіальна, так як точка використовується і в скороченнях. Таблиця скорочень може сильно допомогти під час обробки тексту, щоб уникнути невірної розстановки кордонів пропозицій. У більшості випадків для цього використовуються бібліотеки, так що можете особливо не переживати про деталі реалізації.

Лематизації і стемінг тексту

Зазвичай тексти містять різні граматичні форми одного і того ж слова, а також можуть зустрічатися однокореневі слова. Лематизації і стемінг мають на меті привести все зустрічаються словоформи до однієї, нормальної словникової формі.

Стемінг - це грубий евристичний процес, який відрізає «зайве» від кореня слів, часто це призводить до втрати словотворчих суфіксів.

Лематизації - це більш тонкий процес, який використовує словник і морфологічний аналіз, щоб в результаті привести слово до його канонічної формі - Лемме.

Відмінність в тому, що Стеммер (конкретна реалізація алгоритму стемінг - прім.переводчіка) діє без знання контексту і, відповідно, не розуміє різницю між словами, які мають різний зміст в залежності від частини мови. Однак у Стеммер

є і свої переваги: їх простіше впровадити і вони працюють швидше. Плюс, більш низька «акуратність» може не мати значення в деяких випадках.

1.1.3 Стоп-слова

Стоп-слова - це слова, які викидаються з тексту до / після обробки тексту. Коли ми застосовуємо машинне навчання до текстів, такі слова можуть додати багато шуму, тому необхідно позбавлятися від нерелевантних слів.

Стоп-слова це зазвичай розуміють артиклі, вигуки, сполучники і т.д., які не несуть смислового навантаження. При цьому треба розуміти, що не існує універсального списку стоп-слів, все залежить від конкретної задачі, яка вирішується.

1.1.4 Регулярні вирази

Регулярний вираз (регулярка, regexp, regex) - це послідовність символів, яка визначає шаблон пошуку. Ми можемо використовувати регулярні вирази для додаткового фільтрування нашого тексту. Наприклад, можна прибрати всі символи, які не є словами. У багатьох випадках пунктуація не потрібна і її легко прибрати за допомогою регулярних виразів.

Алгоритми пошуку за регулярними виразами впроваджені у багатьох мовах програмування, але найбільшого поширення набула реалізація з Perl, яка згодом зросла до окремої від мови Perl множини сумісних реалізацій, що зветься PCRE (Perl Compatible Regular Expression).

1.1.5 Мішок слів

Алгоритми машинного навчання не можуть безпосередньо працювати з сирим текстом, тому необхідно конвертувати текст в набори цифр, так звані

вектори. Це називається вилученнням ознак.

Мішок слів - це популярна і проста техніка вилучення ознак, яка використовується при роботі з текстом. Вона описує входження кожного слова в текст.

Щоб використовувати модель, нам потрібно:

- визначити словник відомих слів (токенів);
- вибрати ступінь присутності відомих слів.

Будь-яка інформація про порядок або структуру слів ігнорується. Ось чому це називається мішки слів. Ця модель намагається зрозуміти, чи зустрічається знайоме слово в документі, але не знає, де саме воно зустрічається.

Інтуїція підказує, що схожі тексти мають схожий між собою вміст. Також, завдяки вмісту, ми можемо дізнатися дещо про сенс самого тесту, що досліджується.

1.2 Засоби роботи з текстом

1.2.1 Томіта-парсер

Названий інструмент в честь японського вченого Масару Томіта, автора алгоритму GLR-парсинга (Generalized left-to-right algorithm), на основі якого і був створений Томіта-парсер. Ще в 1984 році він описав імплементацію цього алгоритму, поставивши перед собою завдання ефективно і точно проводити аналіз текстів на природній мові. У деякому роді GLR-алгоритм - це розширена версія алгоритму LR-парсинга. Але LR-алгоритм призначений для аналізу текстів, написаних на досить строго детермінованих мовах програмування, і з природною мовою працювати не може. Томіта вирішив цю проблему шляхом паралелізації стеків, що дозволило розглядати різні трактування тих чи інших ділянок тексту: як тільки виникає можливість різного трактування, стек

розгалужується. Таких послідовних розгалужень може бути кілька, але в процесі аналізу помилкові гілки відкидаються, і результатом стає найбільш довгий ланцюжок. При цьому алгоритм видає результати своєї роботи в режимі реального часу, у міру просування вглиб тексту, інші алгоритми обробки природної мови такою особливістю не володіють. На рисунку наведена схема роботи програми з Томіта-парсером на простому прикладі.

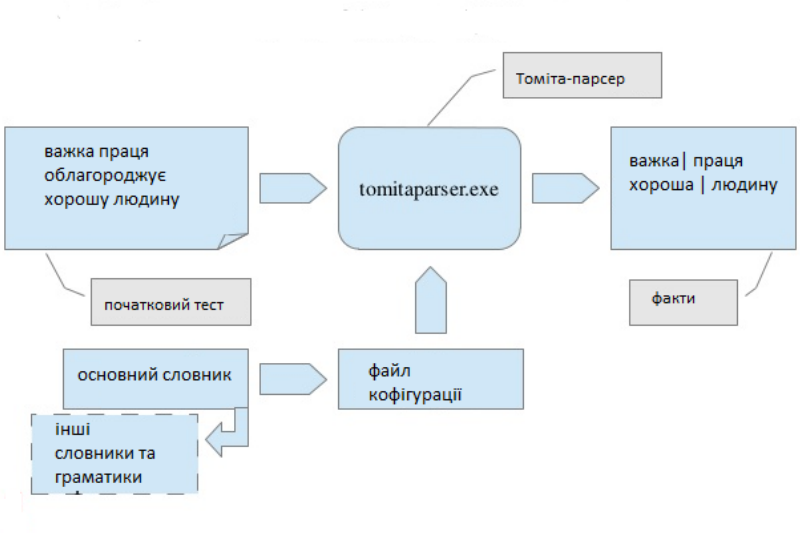


Рисунок 2 - Схема роботи Томіта-парсеру

Однак одного лише алгоритму для повноцінного аналізу тексту і вилучення з нього структурованої інформації недостатньо. Потрібно враховувати морфологію і синтаксис мови оброблюваного тексту, підключити необхідні словники, зрозумілі парсеру. Все це було доступно тільки лінгвістам з досвідом програмування. Томіта-парсер розроблявся спеціально з прицілом на спрощення роботи з алгоритмом. Був складений нескладний синтаксис для створення словників і граматик, продумана робота з морфологією російської та української мов. Тепер при належному завзятості і розумінні пристрою російської мови, розібратися в синтаксисі і підготувати парсер для своїх цілей може практично

будь-хто. Звичайно, більш глибокі пізнання в лінгвістиці і вміння працювати з регулярними виразами буде безумовним плюсом, але вже не обов'язковою умовою.

На сьогоднішній день Томіта використовується в наступних чотирьох сервісах.

- Сервіс «Пошта». Як вже говорилося вище, якщо користувачу цього сервіса лист, в якому в довільній формі пропонується зустрітч, Томіта визначить, де і коли буде проходити ця зустріч, і запропонує внести її в календар. Приблизно так само йде справа з авіаквитками.
- Сервіс «Новини». У цьому сервісі Томіта допомагає автоматично здійснювати географічну прив'язку і угруповання новинних сюжетів. Якщо в замітці згадується назва країни, міста або повна адреса місця, де відбулася описана подія, Томіта виділить цю інформацію і прив'яже замітку до точки на карті.
- Розглянемо сервіс «Авто». Сервіс «Авто» збирає відгуки про автомобілі з різних майданчиків, там же можна залишити свій відгук. Томіта аналізує ці відгуки, оцінює емоційне забарвлення висловлювань про різні характеристики автомобілів. На основі цих даних складається рейтинг характеристик: зовнішній вигляд, ходові якості, салон і комфорт, експлуатація, враження.
- Розглянемо сервіс «Робота». Подібно до «Авто» «Робота» збирає з різних майданчиків оголошення про пошук працівників. Вони також зазвичай складаються в довільній формі. Томіта аналізує ці тексти і виділяє вимоги до кандидатів і умови роботи, формалізує їх, завдяки чому при пошуку роботи користувачі можуть фільтрувати вакансії.

У мінімальній конфігурації парсеру на вході віддається сам аналізований текст, а також словник і граматика. Обсяг словника і складність граматики

залежать від цілей аналізу: вони можуть бути як зовсім маленькими, так і величезними. Файл граматики складається з шаблонів, написаних внутрішньою мовою Томіта-парсера, формальною мовою для парсера. Ці шаблони описують в узагальненому вигляді ланцюжка слів, які можуть зустрітися в тексті. Крім того, граматики визначають, як саме потрібно представляти витягнуті факти в підсумковому висновку.

У словниках містяться ключові слова, які використовуються в процесі аналізу граматиками. Кожна стаття цього словника задає безліч слів і словосполучень, об'єднаних загальною властивістю. Наприклад, «всі міста України». Потім в граматиці можна використовувати властивість «є містом України». Слова або словосполучення можна задавати явно списком, а можна «функціонально», вказавши граматику, яка описує потрібні ланцюжки.

Нижче приведена граматика, яка допомагає вилучити з тексту природною мовою і зв'язати один з одним ім'я режисера, назва і жанр фільму, який він зняв. Також наведено словник, в якому перераховані різні жанри і форми, в яких вони можуть вживатися в тексті.

Прикладом такого інструменту для російської мови можна привести Томіта-парсер, який був розроблений компанією «Яндекс» і в даний час використовується в актуальних для користувача сервісах цієї компанії [4]. Даний інструмент був створений на основі алгоритму GLR-парсінга і дозволяє виділяти з тексту ланцюжки слів (т.зв. факти). Витяг відбувається згідно із зазначеними користувачем правил на мові контекстно-вільних граматик і з використанням словників ключових слів (т.зв. газеттірів).

Граматика для Томіта-парсера має безліч правил мовою контекстно-вільних граматик, які описують структуру виділення ланцюжків. На лістингу 1 наведено приклад найпростішої граматики, за допомогою якої можна витягти всі пари прикметників і іменників. Нижче наведено фрагмент найпростішої

граматика для Томіта-парсера.

```
#encoding “utf-8”

#GRAMMAR_ROOT S

S -> Adj Noun;
```

Газеттіри представляють собою словники ключових слів, розділені на статті або розділи, що визначають множини слів і словосполучень, об'єднаних загальною властивістю.

Результатом роботи Томіта-парсера є список словосполучень з тексту, відповідних зазначеній граматиці і газеттірам. При цьому слова в словосполученні залишаються в узгодженому вигляді і по можливості приводяться до нормальної формі.

1.2.2 Stanford CoreNLP

Stanford CoreNLP - набір бібліотек для розробки засобів аналізу природної мови, що розробляється студентами та співробітниками Стенфордського університету [5]. Даний інструмент був створений в першу чергу для аналізу текстів англійською мовою, але зараз підтримує також китайську, іспанську, німецьку і арабську мови. Stanford CoreNLP - інструмент з відкритим вихідним кодом.

Синтаксичний аналіз текстів дозволяє витягти всі формальні словосполучення, відповідні певним правилам, тобто правилам граматики, проте кількість витягнутих словосполучень з одного тексту може бути занадто велика, що буде ускладнювати аналіз або робити його зовсім неможливим. Синтаксичні аналізатори ніяк не враховують тематику тексту і не здатні відділяти факти, які стосуються заданої теми, від загальновживаних фактів або таких, які стосуються інших тем.

1.3 Вилучення інформації про готелі на основі онтологій

Онтологія – є одним з основних підходів до виділення ключових фраз з неструктурованих текстів на ряду з такими підходами як метод, що опирається на правила та машинне навчання. Онтологією називають формалізоване представлення знань про певну предметну область, тобто середовище, придатне для автоматизованої обробки. Онтологію неодмінно супроводжує деяка концепція цієї області інтересів. Найчастіше ця концепція виражається за допомогою визначення різноманітних базових об'єктів, наприклад, таких як індивідууми, атрибути, процеси тощо, і відношень між ними. Визначення суті цих об'єктів і відношень між самими об’єктами зазвичай називають концептуалізацією.

Таке визначення онтології є узагальнюючим: онтологія — це загальноприйнята і загальнодоступна концептуалізація певної області знань (світу, середовища), яка містить базис для моделювання цієї області знань і визначає протоколи для взаємодії між агентами, які використовують знання з цієї області, і, нарешті, включає домовленості про представлення теоретичних основ даної області знань.

Онтології зазвичай містять класи (поняття), екземпляри цих класів, їхні атрибути (властивості) та значення цих властивостей, а також відношення між класами та екземплярами класів. Крім того, онтологія може містити певні обмеження на використання класів та їх відношень. Об'єкти в онтології можуть мати атрибути. Кожен атрибут має принаймні ім'я й значення, і використовується для зберігання інформації, що специфічна для об'єкта й прив'язана до нього.

Семантична павутина — нова концепція розвитку Всесвітньої павутини і мережі Інтернет, яка створена і впроваджується Консорціумом Всесвітньої павутини. Інші назви — семантичний веб, семантична мережа. Хоча поняття

семантична мережа, яке виникло раніше, породило поняття семантична павутина, їх слід відокремлювати.

Концепція полягає у впровадженні спільних, стандартних форматів даних у Мережі. Для заохочення впровадження семантичного форматування сторінок, пропонується змінювати структуру вже існуючих, не структурованих чи частково-структурованих сторінок у «мережу даних». Створення семантичної Мережі полягає у застосуванні середовища опису ресурсів (RDF).

Семантична павутина — це надбудова над сучасною Всесвітньою павутиною, яка покликана зробити інформацію, що розміщена в мережі, зрозумілішою для комп'ютерів. Відомо, що майже вся інформація в Інтернеті знаходиться в текстовій формі. Не секрет також, що прогрес в галузі обробки людської мови йде дуже повільно. Комп'ютери не можуть сприйняти й осмислити словесну інформацію, розміщену в Інтернеті, і в найближчий час, мабуть, не зможуть. Тоді постає питання — як змусити комп'ютери розуміти зміст розміщеної в мережі інформації і навчити їх користуватися нею? На це питання і покликана відповісти концепція семантичної павутини. Слово «семантична» у цьому випадку означає «осмислена», «зрозуміла».

На сьогодні комп'ютери беруть досить обмежену участь у формуванні й обробці інформації в мережі Інтернет. Функції комп'ютерів, в основному, зводяться до збереження, відображення і пошуку інформації. У той же час створення інформації, її оцінка, класифікація й актуалізація — усе це як і раніше виконує людина. Як включити комп'ютер у ці процеси? Якщо комп'ютер поки не можна навчити розуміти людську мову, то потрібно використовувати мову, що була б зрозумілою комп'ютеру. Тобто, в ідеальному варіанті, вся інформація в Інтернеті повинна розміщуватись двома мовами: людською мовою для людини і комп'ютерною мовою для розуміння комп'ютером. Семантична павутина — це

концепція мережі, в якій кожен ресурс людською мовою був би доповнений описом, зрозумілим комп'ютеру.

В роботі [6] описаний метод вилучення інформації з текстових джерел даних, заснований на використанні семантичної павутини з опорою на онтологію. Автори даної роботи пропонують алгоритм побудови онтологічної бази знань, проводячи частотний аналіз результатів роботи текстового парсеру синтаксичних трійок StanfordCoreNLP [5].

Аналогічним шляхом йдуть автори дослідження [7], розширюючи область застосовності описаних алгоритмів на статті довільної тематики та ілюструючи роботу алгоритмів на прикладі туристичного сектора.

Онтології можуть бути універсальними (в них робиться спроба описати максимально широкий набір об'єктів), галузеві (з інформацією за предметними областями) і вузькоспеціалізовані (призначені для вирішення конкретного завдання). Також можуть застосовуватися онтології об'єктів (бази знань). Найбільш яскравий приклад бази знань - це Вікіпедія.

Отже, на основі онтології, спираючись на контексти і вже наявні списки об'єктів, можна будувати гіпотези стосовно об'єктів і фактам в тексті, а далі верифікувати або відхиляти ці гіпотези. На рисунку 3 зображено скріншот з Вікіпедії з пошуковим запитом. Жирним шрифтом виділені об'єкти, про які хотілося б отримати якусь інформацію. Розглягнма, як онтологія застосовується Вікіпедії. Відправляючи у енциклопедію запити з всіма ключовими словами з тексту, користувач отримує список статей. Першими у цьому списку показані в статті, які стосуються відразу до декількох об'єктів з пошукового запиту користувача.

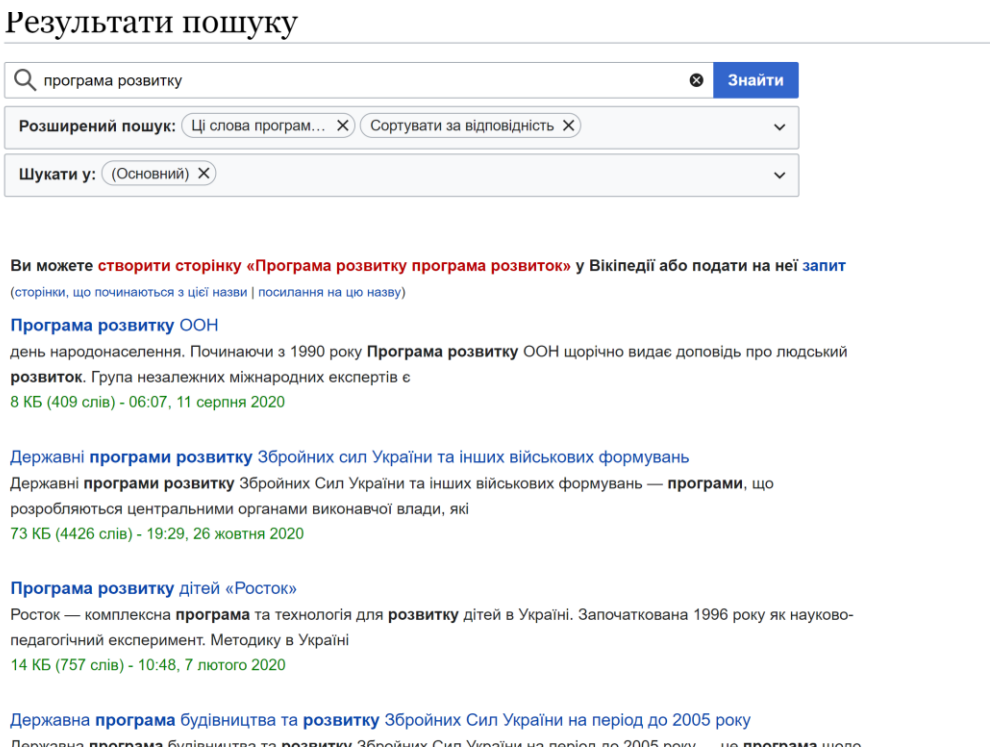


Рисунок 3 – застосування онтології у Вікіпедії

У нашому випадку онтології - це «концептуальні словники», що представляють собою структури, в яких описуються деякі поняття і / або об'єкти, відносини між ними, а також їх характеристики.

1.4 Комерційні сервіси з даними про готелі

У сучасному світі люди інформацію про відпочинок, зокрема про готелі, як правило, дізнаються з Інтернету. Існує безліч сервісів, що спеціалізуються на туристичному напрямку і на яких розміщена інформація про готелі, сайти новин, сайти туристичних операторів, блоги туристичної тематики, сторінки готелів у соціальних мережах тощо

Комерційні сервіси в даній сфері, як правило, використовують закриті технології, подробиці яких не описані у відкритому доступі. Однак, в контексті даної роботи є цікавим для деяких з таких сервісів дати короткий опис функціональності, пов'язаної з побудовою рейтингів по набору фактів.

Розглянемо детальніше сервіс «TrustYou». Сервіс «TrustYou» (http://www.trustyou.com/) надає користувачам коротку інформацію про готелі в вигляді коротких фактів, а також деякий рейтинг, який будується на їх основі. Приклад побудови рейтингів рейтингів готелів по параметрам на основі фактів сервісом TrustYou наведено на рисунку 4.

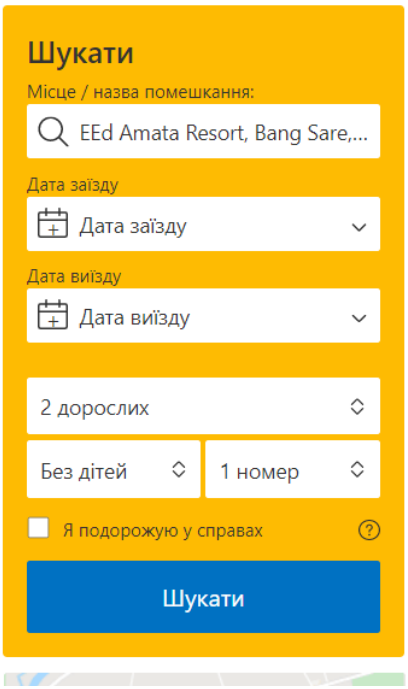
Рисунок 4 - Побудова рейтингів готелів сервісом «TrustYou»

Booking.com. Міжнародний сервіс Booking.com (http://www.booking.com/) також здійснює фільтрацію готельних закладів щодо ключових аспектів розміщення. Крім цього, для кожного конкретного готелю сервіс надає коротку інформацію про особливості об'єкта.

Інвестуючи в технології, які допомагають подорожувати без клопоту, Booking.com пропонує мільйонам гостей приголомшливі варіанти дозвілля, транспортні послуги та неймовірні помешкання: від приватних помешкань до готелів і не тільки. Як найбільша у світі туристична платформа як для відомих

брендів, так і для підприємців різного рівня, Booking.com допомагає власникам помешкань у всьому світі приваблювати гостей і розвивати їхній бізнес.

На рисунках 5 та 6 наведені приклади фільтрів для побудови рекомендацій на сайті «Booking.com».

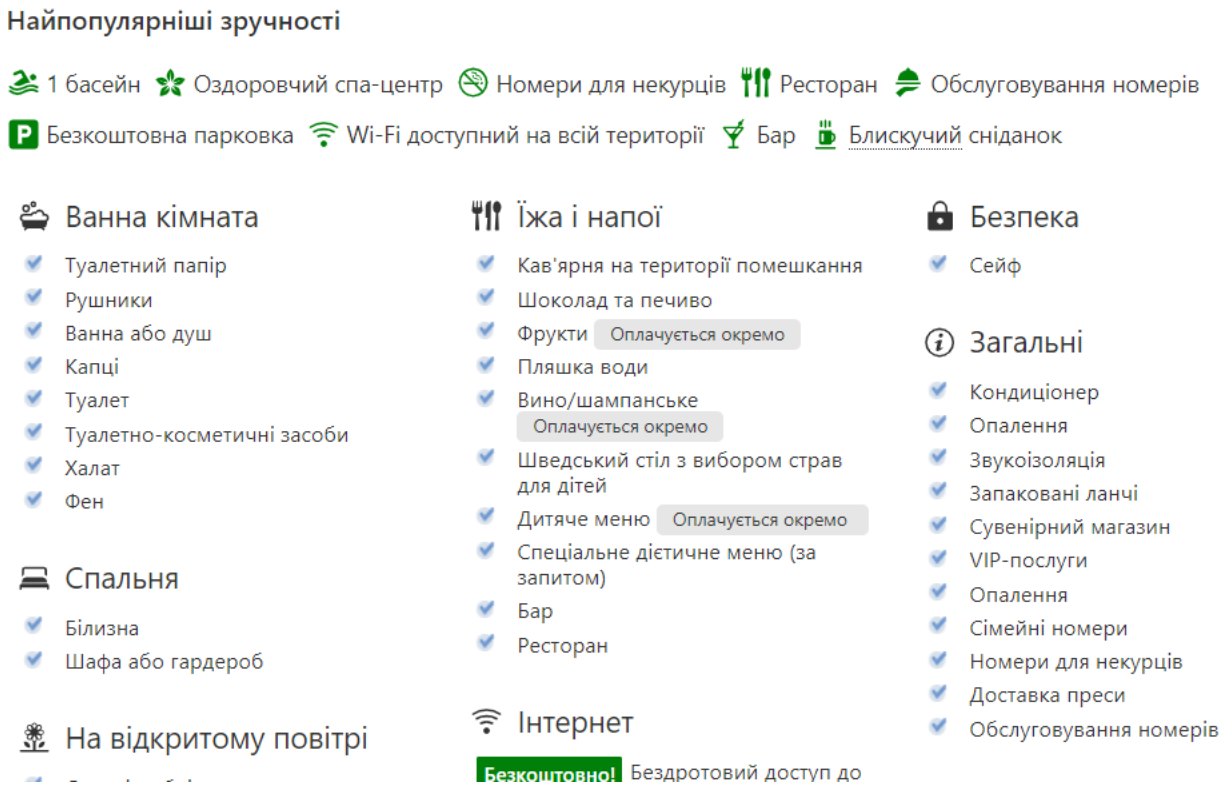


Рисунки 5 - Фільтри сайту Booking.com

На сайт неодноразово подавалися скарги в різних країнах. Так, у Франції сайт звинувачують в ціновому тиску на готелі та порушення антимонопольного законодавства. За повідомленням «The Daily Telegraph», компанія змушує готельних операторів просувати найбільш вигідні для Швейцарії, США і Великобританії.

На 4 квітня 2018 року в базі даних сайту містяться 1 745 254 об'єктів розміщень в 227 країнах, сайт доступний в 42 мовних версіях.

ТопХотелс – ресурс, що спеціалізується на побудові каталогу і рейтингів готелів світу, заснований на думках фахівців туристичного бізнесу та відгуках туристів. Це однин з найбільш повних в україномовному Інтернеті каталог



Рисунки 6 - Опис готелю на сайту Booking.com

готелів. Проект «ТопХотелс» добре відомий на українському туристичному ринку і є лідером за популярністю в своєму сегменті. Більше 10 млн. відпочиваючих щорічно використовують «ТопХотелс» для вибору готелю за даними цього ресурсу, знайомлячись з відгуками інших туристів і читаючи рекомендації професіоналів туристичного бізнесу.

Для туристів «ТопХотелс» надає наведені нижче можливості:

- Вибір для себе кращого варіанту готелю на основі відгуків інших туристів і турагентів;
- Можливість прямо на сайті підібрати тур на основі своїх переваг і бюджету;
- Найкращий майданчик для вибору справжнього професіонала серед турагентів Вашого міста;

- Можливість ділитися своїми найяскравішими враженнями та емоціями від поїздок;
- Пряме спілкування на сайті з готелями і турагентами;

«ТопХотелс» як і багато інших спеціалізованих сервісів мають статті з описом готелів, зокрема їх характеристик та особливостей для формування в уяві користувача образу готелю якомога ближчого до реального стану речей у цьому готелі.

«ТопХотелс» надає користувачеві можливість встановлювати деякі фільтри як наведено нижче на рисунку 7.

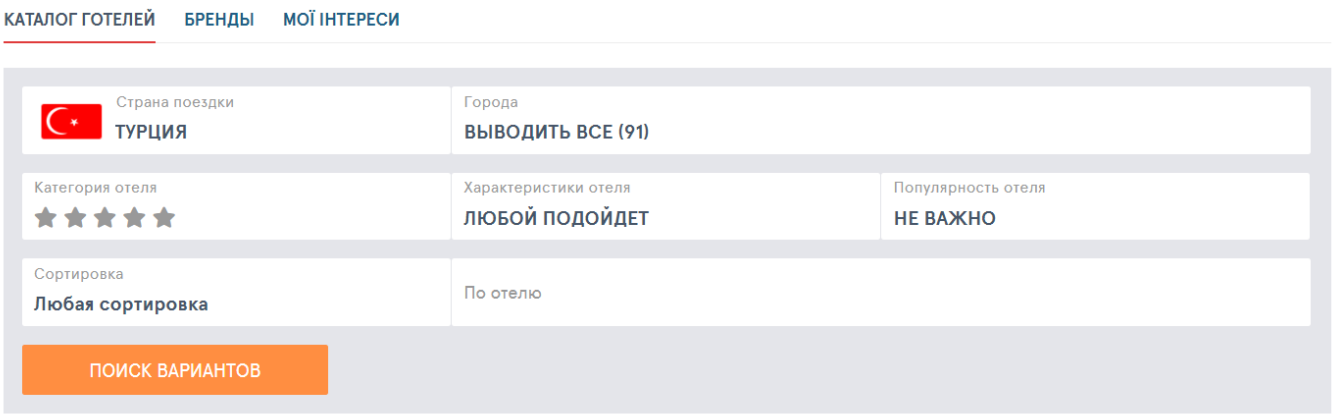


Рисунок 7 - Приклад організації фільтрів «ТопХотелс»

Різні ресурси організовують систему рекомендацій та фільтрів по різному, але є деякі спільні риси. Більшість з ресурсів намагаються зробити фільтри за такими ознаками як країна регіон, середня оцінка готелю, тип номеру тощо. Це пояснюється тим, що більшість з цих фактів є у офіційному описі готелю. Але також ресурси показують користувачам короткий опис готелю як набір фактів про нього.

ВИСНОВКИ ДО РОЗДІЛУ 1

Один з видів документів, для яких необхідно виділення ключових фраз є статті та відгуки про готелі. Завдання складання ключових фраз для текстів туристичної галузі ручним способом є трудомісткою і забирає багато часу у менеджерів відповідних ресурсів. Рішенням даної задачі стає автоматичне вилучення ключових фраз.

Оскільки ключові фрази відображають основну ідею документа, від вилучення правильних ключових фраз залежить ефективність додатків по обробці природних мов.

У першому розділі було розглянуто основні ресурси для пошуку та підбору туристичних послуг, зокрема готелів. Для кожного з цих ресурсів («http://www.trustyou.com/», «http://www.booking.com/» та «http://tophotels») було наведено приклади рекомендаційних систем. Було виявлено загальні риси для цих ресурсів.

Витяг фактів з текстів написаних природною мовою є зовсім нетривіальним завданням в світі розробки ПЗ. Томіта-парсер надає можливості для елегантного вирішення цієї непростої проблеми. Проте, все одно вкрай складно написати універсальну граматику, яка розбереться у всіляких варіантах вхідних послідовностей. Тому робота з природними мовами як і раніше залишається досить складною. У першому розділі наведені приклади способів роботи з текстами природною мовою і продемонстровані недоліки цих методів.

2. АЛГОРИТМИ ВИЛУЧЕННЯ КЛЮЧОВИХ ФАКТІВ

Розглянувши існуючі рішення у першому розділі, легко зрозуміти, що кожен з них має певні недоліки. Для створення рекомендацій для ресурсів з великою кількістю неструктурованих текстів слід проаналізувати переваги та недоліки вже існуючих алгоритмів та перейняти якомога більше позитивних характеристик у створення нового алгоритму.

Основними особливостями способу, що розробляється, слід зробити підходящими для роботи з великою кількістю неструктурованого тексту та з великою кількістю фактів.

Для способу вилучення основних ключових фраз з неструктурованих текстів, якими є відгуки та статті про готелі, важливим є виділять ключові фрази за тематикою тексті. Також важливою рисою такого алгоритму є вилучення об’єктивних фактів, на які не впливає суб’єктивна думка про даний об’єкт автора тексту.

2.1 Алгоритм, заснований на частотності фактів

Один з алгоритмів, який взято за основу способу, що розробляється – алгоритм, заснований на частоті фактів.

Алгоритм, заснований на частотності фактів – підхід спрямований на скорочення множини фактів, витягнутих з допомогою обраного синтаксичного парсеру (в даному випадку Томіта-парсера). Цінність скороченої підмножини фактів полягає в тому, що вона допомагає описати предметну область статей, у даному випадку туристичну сферу і готелі зокрема, не використовуючи «зайвих» словосполучень, тобто синтаксичних словосполучень, що не відносяться до головної теми статті. Алгоритм заснований на підрахунку частотності словосполучень і таким чином служить для виділення підмножини фактів, що

описують характеристики, загальні для всіх об'єктів предметної області. Наприклад, для готелів в якості прикладів таких характеристик можна привести ознаки «пляж», «Інтернет», «сервіс».

2.1.1 Особливості реалізації

Перед початком роботи розробленого способу необхідно вилучити ключові факти з неструктурованих текстів. Для вилучення фактів було обрано використання Томіта-парсеру. У якості словників, які використовує цей парсер для вилучення фактів буде використано розроблені тематичні словники. Особливості розробки таких словників описані нижче.

Розроблений спосіб вилучення основних ключових фраз складається з кроків які наведені нижче.

1. Відфільтрувати витягнуті синтаксичні факти з використанням спеціально складених словників, що містять тільки ключові характеристики обраної області (наприклад, «пляж», «Інтернет» і ін.). Даний крок опціональний тому, що якщо не фільтрувати характеристики по словниках, статистика буде вважатися за всіма витягнуті словосполученням. Незважаючи на це, в топ найбільш уживаних фактів вийдуть словосполучення, найчастіше з'являються в статтях обраної тематики (в даному випадку це статті та відгуки про готелі).

2. Підрахувати статистику появи залишилися фактів по всій текстової базі навчальної множини.

3. Відсортувати факти спаданням частоти.

4. Взяти перші N відсотків фактів за шукану підмножину словосполучень S.

5. На тестовій множині оцінити якість вилучення ключових фактів предметної області з використанням підмножини S. Схема оцінки якості алгоритму описана нижче в п. 2.3.1.

6. Змінюючи параметр N і повторюючи кроки 1-5 даного алгоритму, домогтися бажаного якості вилучення фактів.

2.1.2 Практичне застосування: побудова рейтингів готелів

Оскільки мета даного способу полягає у виділенні фактів, які відносяться до характеристик, спільних для всіх об'єктів обраної предметної області, представляється можливим подальша побудова рейтингів об'єктів на основі витягнутих словосполучень.

Під час вилучення факту *f* з тексту інформацією про цей готель *h* визначимо вагу f для об'єкта *h* як відношення частоти появи *f* у всіх текстах про об'єкт *h* до частоти появи *f* по всій текстової базі (тобто для всіх об'єктів). У разі появи в результаті суперечливих фактів пропонується порівнювати їх ваги для вибору більш важливого. В таблиці 1 представлений приклад витягнутих фактів з вагами для одного конкретного готелю.

Таблиця 1 - Приклад частотних фактів для конкретного готеля

Amata Resort Пхукет Таїланд				
параметр	визначення	правильно	вага для готелю	частота по базі
Інтернет	платний	false	0.02	0.25
сніданок	Безкоштов-ний	false	0.02	0.67
wi-fi	Безкоштов-ний	true	0.08	0.68

На даному прикладі видно, що факт «wifi: безкоштовний» має більшу вагу, ніж факт «інтернет: платний», як для конкретного готелю, так і по всій базі

витягнутих фактів в цілому. Таким чином, при побудові рейтингів, а також при оцінці якості вилучення фактів миможемо враховувати тільки факти з найбільшою вагою для запобігання протиріч.

2.2 Алгоритм, заснований на статистичній мірі TFIDF

TFIDF складається з двох компонентів: термінова частота (частота слов у документах) та обернена частота документа (інверсія частоти документа). Підрахувати ці величини можна за наведеними нижче формулами:

$$TF - IDF = TF * IDF$$

$$TF_{token_i} = \frac{n_i}{N_i}$$

$$IDF_{token} = log\frac{p}{P}$$

де n_i - скільки раз зустрічається токен в і-тому документі,

Ni - загальна кількість токенів у *i*-тому документі,

p - кількість документів, у яких зустрічається токен,

P - загальна кількість документів.

У кінцевому навчанні, TF-IDF - це похідна TF на IDF

Підхід явного вилучення ознак хороший тим, що зазвичай отримані ознаки легко інтерпретувати і зрозуміти. Неявні ознаки більш важкі для розуміння, однак такий підхід в окремих випадках здатний визначати складні властивості коду, такі як, наприклад, семантичні залежності.

Даний підхід спрямований на витяг фактів, що описують особливості конкретного об'єкта на противагу загальним характеристикам предметної області. В основі запропонованого способу лежить статистична міра TFIDF [8], що дозволяє оцінити важливість слова в контексті документа, який є частиною деякої колекції документів. Важливість слова для документа пропорційна частоті вживання цього слова в даному документі, і обернено пропорційна частоті

вживання слова у всіх інших документах з колекції.

У контексті даної роботи в ролі слів виступають синтаксичні факти, витягнуті з допомогою Томіта-парсера, а в якості одного документа приймається об'єднання всіх текстів, що відносяться до одного готелю.

2.2.1 Особливості реалізації

Позначення. Приймемо наступні позначення. Articles (h) - всі статті, які стосуються готелю h. Articles (H) - всі статті, які стосуються готелям з поточної бази даних [9].

Розроблений спосіб. Даний спосіб складається з наступних кроків.

1. Для даного готелю *h* взяти всі тексти, що відносяться до даного об'єкта, витягти всі синтаксичні факти за допомогою парсера.

2. Для кожного синтаксичного факту *f* обчислити величину
R (f) = tf (f, Articles (h)) * idf (f, Articles (H)).

3. Відфільтрувати всі факти з величиною R> T, де T - обраний поріг. Прийняти отриману підмножину за шукану множину фактів.

4. Оцінити якість вилучення характерних фактів. Варто зауважити, що в даному випадку не представляється можливим виділити окремі етапи навчання і тестування моделі. Для інтегрування описаного підходу в рекомендаційні системи, що працюють в реальному часі, необхідно здійснити попередній підрахунок фактів даного виду для обраної множини об'єктів.

5. Змінюючи параметр T і повторюючи кроки 1-4 алгоритму, домогтися бажаної якості вилучення фактів.

2.2.2 Практичне застосування: отримання нестандартної важливої інформації

Специфіка метрики TF-IDF, на відміну від інших аналогічних, дозволяє виділяти особливості, характерні тільки для одного об'єкта колекції і таким чином виділити даний об'єкт серед всіх інших елементів [10]. В контексті коротких характеристик готелів, в ролі яких в даній роботі розглядаються факти, розроблена метрика надає можливість отримувати відмінні риси окремих готелів, які були б цікаві користувачам, але не вийшли б у топ частотних словосполучень за підсумками першого алгоритму, описаного в п. 2.2.1.

В таблиці 2 наведені приклади вилучення подібних характеристик, отримані в результаті роботи даного алгоритму.

Таблиця 2 - Приклад специфичніх фактів для готелів.

текст	параметр	визначення	правильність
«По пляжу повзають черепахи»	черепаха	повзать	true
«Після цього басейна у мене зелёне волосся!»	волося	зелений	true

2.3 Перевірка ефективності алгоритмів

2.3.1 Оцінка алгоритму, заснованого на частотності фактів

Оцінка алгоритму, заснованого на частотності фактів, проводилася з використанням модифікації стандартної метрики «точність» 13. Також зважаючи на велику кількість вихідних даних і витягнутих фактів було використано випадкове семплірування: були побудовані 5 випадкових вибірок

по 30 унікальних ідентифікаторів готелів, і оцінювалася точність вилучення фактів з відповідних цим ідентифікаторів статей [11].

Всі витягнуті ключові факти були розбиті на 4 типи, що відповідають поняттям True Positive, True Negative, False Positive і False Negative, які зазвичай використовуються для оцінки якості бінарної класифікації. Оцінка істинності факту проводилася вручну з використанням підтверджених джерел: офіційний сайт готелю і опис готелю на сайтах Booking.com і TopHotels. Далі за наведеною нижче формулою була обчислена метрика ACC (від англ. Accuracy - «Точність») окремо для кожної випадкової вибірки.

ACC = (TP + TN) / (TP + TN + FP + FN)

В результаті було отримано, що з імовірністю 95% довірчий інтервал (0.74, 0.89) містить ACC алгоритму на випадковою вибіркою. ACC п'яти побудованих вибірок потрапили в даний довірчий інтервал.

Однією з труднощів, яка проявилася в процесі оцінки, виявилася велика кількість суб'єктивних характеристик готелю, коректність яких не представляється можливим оцінити незалежно [12]. Прикладами фактів з подібними характеристиками є такі словосполучення, як «відпочинок найкращий», «пляж: шикарний» и т.п. Ця проблема була вирішена фільтрацією словосполучень, що містять визначення, які виражені якісними прикметниками, які позначають відношення до об'єкту (хороший, розкішний, шикарний і т.п.).

2.3.2 Оцінка алгоритму, заснованого на TF-IDF

У разі алгоритму, в основі якого лежить модифікація метрики TF-IDF, не представляється можливість оцінити об'єктивні величини. Найчастіше факти, витягнуті за даною схемою, не підлягають об'єктивній «Віддаленій» оцінці. Однак, була виявлена наведена далі залежність: в разі, якщо в побудованому

рейтингу фактів перші 2-3 місця займали факти, вага яких більш ніж в 4 рази перевищує вагу наступних в топі фактів, то з ймовірністю 89% дані факти передають особливості конкретного готелю.

2.3.3 Оцінка інформативності

Оцінку інформативності фраз можна обчислити за такими характеристиками як частота слів і розташування слів в документі. Частота слів містить кількість входжень слова або фрази в досліджуваному документі. Чим більше входжень досліджуваного слова в документ, тим вище його інформативність [13]. Але необхідно фільтрувати часто зустрічаються слова (стоп-слова), які не містять будь-якої інформації про документ.

Розташування слів в документі так само враховується, найбільш інформативні фрази зустрічаються, як правило, на початку документа, в анотації і в заголовку.

Найпоширеніша міра для розрахунку інформативності термінів в документі є TF-IDF. Вага терміна пропорційна кількості вживань даного терміна в документі, і обернено пропорційна частоті вживання в інших документах колекції. Особливістю цього заходу є те, що при зміні кількості документів в колекції необхідно перераховувати частоти всіх термінів.

$$TF - IDF = TF * IDF$$

TF (Term Frequency) - частота фрази в аналізованому тексті, ставлення числа входження фрази до загальної кількості фраз у документі. Розрахувати частоту, з якою зустрічається дана фраза у множині текстів, що досліджуються можна за формулою.

$$TF = \frac{n_i}{\sum_k n_k}$$

IDF (Inverse Document Frequency) - інвертована частота документа - зворотна частота, з якою термін зустрічається в інших документах колекції [14]. Розрахувати інвертовану частоту фрази у тесті можна за формулою:

$$IDF = log\frac{N}{df}$$

де:

N - загальна кількість документів в колекції (корпусі);

df - кількість документів, що містять термін.

Вибір підстави логарифма не має значення так як не впливає на співвідношення ваг термінів.

2.3.4 Оцінка ефективності

Не менш важливою частиною завдання є оцінка ефективності знайденого рішення. Ефективність оцінюється відносно до множини автоматично знайдених ключових фраз в текстах по відношенню до заздалегідь відомих ключових фактів.

Точність (Precision) - відношення числа релевантних ключових фактів знайдених автоматично, до загальної кількості знайдених ключових фатів в текстах, що обробляються. Розрахувати цю точність можна за формулою:

$$P = \frac{|Trel \cap Tretr|}{|Tretr|}$$

де:

P - точність;

Trel - безліч релевантних термінів;

Tretr - безліч знайдених термінів.

Повнота (Recall) - це відношення числа релевантних ключових фактів, що були знайдені автоматично, до загальної кількості релевантних ключових фраз

в документі [15]. Розрахувати повноту можна за формулою:

$$R = \frac{|Trelf \cap Trel|}{|Trelf|}$$

де:

R - повнота;

Trelf - безліч релевантних термінів містяться в документі;

Trel - безліч знайдених релевантних термінів.

F-міра (F-score, F-measure) - об'єднання точності і повноти в одній усередненої величиною, визначається як зважене гармонійне середнє точності і повноти.

$$F = \frac{1}{\alpha\frac{1}{p} + (1-\alpha)\frac{1}{R}}, \quad \alpha \in [0,1]$$

де:

F - міра;

P - точність;

R - повнота.

2.4 Дані для експерименту

Для проведення експерименту щодо перевірки ефективності розробленого способу необхідно отримати множину відгуків та статей про готелі.

Дані для експериментів були зібрані на основі реального досвіду користувачів пошукових систем в такий спосіб. Був обраний топ найбільш популярних готелів, тобто готелів, назви яких найчастіше фігурують в запитах користувачів до пошукової системи. Для даних готелів були вивантажені два типи документів: так звані статті як видача пошукової системи на запит з назвою готелю та так звані відгуки як результат аналогічного пошуку з

маркером «відгуки» в тілі запиту [16]. За допомогою Томіта-парсера з вивантажених текстів були вилучені всі синтаксичні біграми, факти зо складаються з двох частин.

Для вилучення цих фактів слід використати відповідну граматику для парсера і великі словники, складені на основі аналогічних словників, розроблених іншими мовами та перекладені на українську мову за допомогою машинного перекладу.

2.4.1 Попередня обробка статей

Спочатку неструктурований текст, що аналізується, може містити деякі «зайві» елементи (таблиці, примітки автора), які не сприяють вилученню ключових фраз, тому необхідно виконати кілька кроків попередньої обробки. Попередня обробка включає усунення приміток і позначок автора.

Визначення заголовків у статей є корисним, проте зустрічаються заголовки складаються з кількох рядків [17]. Для вирішення цієї проблеми використовується Google API - програмний інтерфейс популярної пошукової системи Google. Необхідно відправити веб-запит до цього інтерфейсу з першим рядком документа і з 10 перших відповідей вибрати один, який і буде найбільш імовірним заголовком. Цей заголовок додається в початок документа, а всі рядки до анотації опускаються.

Рядки, які швидше за все не містять значущу інформацію, так само виключаються з документа. Ці рядки визначаються згідно зі статистичними даних їх зовнішніх характеристик (наприклад, середня довжина рядка і її відхилення) і регулярними виразами [18]. Абзац і межа пропозиції знаходяться за допомогою алгоритму, заснованого на правилах, також проводяться граматичне (POS, Part-of- speech tagging) і синтаксичне тегування кожного речення за допомогою Stanford parser.

Після отримання синтаксично обробленої пропозиції стає можливим вилучити з усієї множини текстів, що обробляються, «кандидатів» у ключові фраз. Послідовності від одного до чотирьох слів, які не починаються і не закінчуються на стоп-слова, а так само містять теги тільки прикметника, іменника або дієслова і визначаються як кандидати ключових фраз. Слова, з яких складаються кандидати у ключові фрази, необхідно перетворити в універсальну форму для збереження.

ВИСНОВКИ ДО РОЗДІЛУ 2

У другому розділі роботи було наведено та проаналізовано існуючі методи роботи з ключовими фразами вилученими з тексту. Для кожного з методів було знайдено позитивні та негативні елементи.

На основі вже існуючих алгоритмів було створено та описано новий алгоритм вилучення ключових фраз з неструктурованих текстів. Особливостями розробленого методу є використання тематичних словників, скорочення множини слів у словниках за принципом виключення емоційно забарвлених слів, встановлення порогу якості вилучених фактів.

Також у даному розділі були описані методи оцінки інформативності вилучених фактів на основі всієї множини фактів з даної галузі. У роботі запропоновані методи оцінки ефективності вилучення ключових фактів з неструктурованих текстів.

3 РЕАЛІЗАЦІЯ РОЗРОБЛЕНОГО АЛГОРИТМУ

3.1 Загальна структура розробленої програми

Для демонстрації роботи розробленого способу розроблено алгоритм його реалізації, а також створена програма з використанням цього алгоритму. Особливістю розробленої програми є виділення ключових фактів з неструктурованих текстів, а саме відгуків та статей про готелі.

Програма складається з наступних модулів:

- база даних, яка зберігає:
  - множину готелів, а саме їх ідентифікатори назви та місце положення;
  - відгуки про готелі, а саме ідентифікатор даного відгуку, текст відгуку, дату та автора, ресурс, з якого вилучено даний відгук про готель;
  - статті про готелі, а саме ідентифікатор статті, текст даної статті, ресурс, з якого вилучено дану статтю, дату вилучена, ім’я автора статті;
  - словник фраз, а саме множину слів та словосполучень, яка може описувати надання готельних або інших послуг у туристичній сфері;
  - множину фактів про готелі, а саме ідентифікатор готелю, ідентифікатор факту, показник правдивості цього факту щодо даного конкретного готелю;
- парсер – це елемент програми, що вилучає ключові факти з текстів, тобто елемент програми, що розбирає неструктурований тест за граматичними основами та виділяє пари слів, факти;

- аналізатор фактів про готелі – це елемент програми, що аналізує вилучені за допомогою парсеру факти, будує статистичне порівняння фактів, формує частотну характеристику даного факту та помічає позитивну, негативну чи нейтральну характеристику факту.

Структура проєкту складається з таких папок:

- data – папка, яка містить факти про готелі, структуровані за їх характеристиками та способом вилучення;
- data_results_frequency – папка, що містить дані про готелі, а саме співвідношення фактів, їх характеристики, частоту вживання у статтях та відгуках про даний готель та приблизну достовірність;
- data_results_tfidf – папка, яка містить дані про готелі, а саме співвідношення вилучених за допомогою парсеру фактів, характеристики цих фактів, частоту вживання у статтях та відгуках про готелі та приблизну достовірність вилучених за допомогою алгоритму TFIDF;
- machine_translation – папка, що містить файли з кодом, які використовуються для машиного перекладу з однієї мови на іншу словників та деяких відгуків;
- other - папка, що містить статичні файли для роботи програми;
    - slovariki_tomita – папка, яка містять словники для виділення ключових фактів за тематикою туристичного бізнесу;
    - allResortsCountries12MonthsWithTourMarker.xlsx – перелік країн та міст за кількістю готелів;
    - countriesResortsStatFromQueries.xlsx – статистика кількості запитів користувачами мережі Інтернет за країнами та курортами;
    - hotelQueryStatistics.xlsx – статистика кількості запитів користувачами мережі Інтернет щодо даних про готель;

- src – папка, яка містить основні файли з кодм програми для роботи зі словниками та тестами;
    - HotelsFact.java – файл, у якому відбувається перевірка фактів щодо коректності їх задання та відповідності до правил утворення фраз;
    - HotelsFactsAnalyzer.java – файл, у якому описано вилучення фактів про готель з бази даних та аналіз щодо позитивної або негативної оцінки цих фактів;
    - HotelFactsFrequencyAnalyzer.java – файл, у якому описано метод аналізу частотності фактів вилучених зі статей та відгуків за допомогою парсеру;
    - HotelsFactTFIDFAnalyzer.java – файл, у якому описано вилучення фактів за допомогою алгоритму TFIDF та описано аналіз вилучених за допомогою цього алгоритму фактів щодо їх частотності та достовірноті;
    - TomitaFacts.java – файл, у якому описана робота з фактами за допомогою Томіта-парсеру;
    - TomitaFactsGetter.java - файл, у якому описано вилучення фактів за допомогою Томіта-парсеру;
    - Utals.java – файл, у якому описана взаємодія програми зі статистичними даними у вигляді словників та описано підключення модулів та файлів.

Розглянемо ці елементи розробленої програми, їх особливості реалізації та їх функції у всьому проекті, детальніше у наступних пунктах даного розділу.

3.2 Аналізатор

Аналізатор – це частина програми, яка відповідає за наведені нижче операції з даними:

- перебір вилучених за допомогою парсеру фактів про готелі зі статей та відгуків;
- аналіз фактів щодо їх відповідності до загальної тематики тексту, що аналізується, у даному випадку тематикою є туристична сфера, зокрема готелі;
- відсіювання фактів, що виражають суб'єктивну думку оповідача або не об'єктивну оцінку стану готелю або деяких його елементів за словником відповідних слів;
- встановлення частотності виявлення даних фраз у статтях та відгуках про готель;
- виділення множини фактів, частота яких більша за визначений поріг частоти вилучених фактів;

Нижче наведені фрагменти коду, що відповідає за додавання нових факті до множини фактів про готель.

```
public void addFacts(Set<Integer> hotelIds) {
        int i = 0;

        for (int hotelId: hotelIds) {
            for (Map.Entry<HotelFact, Integer> hotelFact :
getFactsForHotel(hotelId, false, null).entrySet()) {
                HotelFact fact = hotelFact.getKey();
                int freq = hotelFact.getValue();
                _facts.put(
                        fact,
                        _facts.containsKey(fact) ? _facts.get(fact) + freq :
freq
                );
            }
        }
    }
```

Нижче наведені фрагменти коду, що відповідає за вилучення фактів з найбільшою частотою з множини фактів про готель.

```
public void addTopFacts() {

        List<HotelFact> factsList = new ArrayList<>(_facts.keySet());
        Collections.sort(factsList, new Utils.FactComparator<>(_facts));
        for (int i = 0; i < factsList.size() *
Utils.HOTEL_FACTS_FREQUENCY_THRESHOLD; i++) {
            HotelFact hotelFact = factsList.get(i);
            _topFacts.put(hotelFact, _facts.get(hotelFact));

        }

    }
```

Нижче наведений фрагмент коду, що визначає правдивість визначеного факту та чи є він позитивним або негативним.

```
    public void analize() {

        _topFacts.clear();
        _facts.clear();
        System.out.println("De-serializing top facts...");
        try {
            ObjectInputStream ois = new ObjectInputStream(new
FileInputStream(Utils.RESORT_FACTS_STATS_FREQUENCY));
            while (true) {
                HotelFact k = (HotelFact) ois.readObject();
                int v = ois.readInt();
                _topFacts.put(k, v);
            }
        } catch (EOFException e) {
            // skip
        } catch (Exception e) {
            e.printStackTrace();
        }

        Set<Integer> hIds = Utils.readHotelIds(Utils.RESORT_IDS_ALL);
                try {
            BufferedWriter out = new BufferedWriter(new
FileWriter(Utils.RESORT_FACTS_STATS_FREQUENCY_TEST));
            int resCnt = 0;
            for (int hId : hIds) {
                Map<HotelFact, Integer> facts = getFactsForHotel(hId, true,
null);
                int cnt = 0;
```

```
                for (Map.Entry<HotelFact, Integer> hotelFact : facts.entrySet())
{
                    cnt += hotelFact.getValue();
                }
                if (cnt == 0) { cnt = 1; }
                for (Map.Entry<HotelFact, Integer> hotelFact : facts.entrySet())
{
                    HotelFact fact = hotelFact.getKey();
                    HotelFact factNoSaveNegative = new HotelFact(fact.id(),
fact.param(), fact.opinion(), fact.isNegative(), false);
                    double rateHotel = ((double) hotelFact.getValue()) / cnt;
                    double rateDB = ((double) _topFacts.get(factNoSaveNegative))
/ Utils.RESORT_FACTS;
                    double rate = rateHotel / rateDB;
                    if (hotelFact.getValue() > 1 && rate > 1.0) {
                        out.write(hId + "\t" + factNoSaveNegative.toString() +
"\t" + fact.isNegative() + "\t" + rateHotel + "\t" + rateDB);
                        out.newLine();
                    }
                }

            }
            out.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
public Map<String, Double> tfidfForHotel(int hId) {
        Map<String, Double> res = new HashMap<>();
        Map<HotelFact, Integer> factToFrequency = new HashMap<>();
        int factCount = 0;
        if (_connection != null) {
            try {
                _reviewIdSt.setInt(1, hId);
                ResultSet reviewIdRs = _reviewIdSt.executeQuery();
                while (reviewIdRs.next()) {
                    _factIdSt.setInt(1, reviewIdRs.getInt("review_id"));
                    ResultSet factIdRs = _factIdSt.executeQuery();
                    while (factIdRs.next()) {
                        _factSt.setInt(1, factIdRs.getInt("fact_id"));
                        ResultSet factRs = _factSt.executeQuery();
                        while (factRs.next()) {
                            int id = factRs.getInt("id");
                            String param =
factRs.getString("param").toLowerCase();
                            String opinion =
factRs.getString("opinion").toLowerCase();
                            boolean isNegative =
factRs.getBoolean("is_negative");
```

```
                            HotelFact fact = new HotelFact(id, param, opinion,
isNegative, false);
                            factToFrequency.put(fact,
factToFrequency.containsKey(fact) ? factToFrequency.get(fact) + 1 : 1);
                            factCount++;
                        }
                    }
                }

                for (Map.Entry<HotelFact, Integer> ff :
factToFrequency.entrySet()) {
                    double tf = factToFrequency.get(ff.getKey()) * 1.0 /
factCount;
                    _hotelCountSt.setInt(1, ff.getKey().id());
                    ResultSet hotelCountRs = _hotelCountSt.executeQuery();
                    int hotelCount = 0;
                    if (hotelCountRs.next()) {
                        hotelCount = hotelCountRs.getInt(1);
                    }
                    double idf = Math.log(_allHotelsNumber * 1.0 / hotelCount);
                    res.put(ff.getKey().toString(), tf * idf);
                }
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }

        return res;
    }
```

Також розглянемо аналіз фактів за допомогою алгоритму TFIDF. Аналіз за допомогою цього алгоритму у роботі використано для порівння розробленого алгоритму та вже існуючих аналогів. Нижче навежено фрагмент коду, що описує роботу цього алгоритму.

```
public Map<String, Double> tfidfForHotel(int hId) {
        Map<String, Double> res = new HashMap<>();
        Map<HotelFact, Integer> factToFrequency = new HashMap<>();
        int factCount = 0;
        if (_connection != null) {
            try {
                _reviewIdSt.setInt(1, hId);
                ResultSet reviewIdRs = _reviewIdSt.executeQuery();
                while (reviewIdRs.next()) {
                    _factIdSt.setInt(1, reviewIdRs.getInt("review_id"));
```

```
                    ResultSet factIdRs = _factIdSt.executeQuery();
                    while (factIdRs.next()) {
                        _factSt.setInt(1, factIdRs.getInt("fact_id"));
                        ResultSet factRs = _factSt.executeQuery();
                        while (factRs.next()) {
                            int id = factRs.getInt("id");
                            String param =
factRs.getString("param").toLowerCase();
                            String opinion =
factRs.getString("opinion").toLowerCase();
                            boolean isNegative =
factRs.getBoolean("is_negative");
                            HotelFact fact = new HotelFact(id, param, opinion,
isNegative, false);
                            factToFrequency.put(fact,
factToFrequency.containsKey(fact) ? factToFrequency.get(fact) + 1 : 1);
                            factCount++;
                        }
                    }
                }

                for (Map.Entry<HotelFact, Integer> ff :
factToFrequency.entrySet()) {
                    double tf = factToFrequency.get(ff.getKey()) * 1.0 /
factCount;
                    _hotelCountSt.setInt(1, ff.getKey().id());
                    ResultSet hotelCountRs = _hotelCountSt.executeQuery();
                    int hotelCount = 0;
                    if (hotelCountRs.next()) {
                        hotelCount = hotelCountRs.getInt(1);
                    }
                    double idf = Math.log(_allHotelsNumber * 1.0 / hotelCount);
                    res.put(ff.getKey().toString(), tf * idf);
                }
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }

        return res;
    }

    public static void main(String[] args) {
        HotelFactsTFIDFAnalyzer analyzer = new HotelFactsTFIDFAnalyzer();
        Set<Integer> hIds = Utils.readHotelIds(Utils.HOTEL_IDS_N);


        try {
            BufferedWriter out = new BufferedWriter(new
```

```
FileWriter(Utils.HOTEL_FACTS_TFIDF_N_NO_DICT));
            int i = 0;
            for (int hId : hIds) {
                System.out.println("Doing for hotel #" + ++i + " ID: " + hId);
                out.newLine();
                out.write("===============");
                out.newLine();
                out.write("ID: " + hId);
                out.newLine();
                Map<String, Double> m = analyzer.tfidfForHotel(hId);
                String[] fs = m.keySet().toArray(new String[m.keySet().size()]);
                Arrays.sort(fs, new Utils.FactComparator(m));                for
(String f : fs) {
                    out.write(f + " / " + m.get(f));
                    out.newLine();
                }
                out.write("===============");
                out.newLine();
                System.out.println("Done for hotel #" + i + " ID: " + hId);
            }
            out.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
```

3.3 Парсер

Парсер – це частина програми, що вилучає ключові факти з текстів, тобто розбирає неструктурований тест за граматичними основами та виділяє пари слів, факти. Основою парсеру, що використано, є Томіта-парсер.

Парсер розбиває текст статей та відгукав про готелі на лексеми згідно з граматичними правилами та словниками тематичних слів для туристичної сфери. Нижче наведено фрагмент коду, що вилучає такі факти.

```
public class TomitaFactsGetter {
    public TomitaFacts getTomitaFacts() {
        return new TomitaFacts(getTomitaFacts(true), getTomitaFacts(false));
    }

    private Map<String, Set<String>> getTomitaFacts(boolean positive) {
        Map<String, Set<String>> facts = new HashMap<>();
```

```
        String tomita = positive ? _tomitaPositive : _tomitaNegative;

        try {
            BufferedReader in  = new BufferedReader(new FileReader(tomita));
            String str;
            while ((str = in.readLine()) != null) {
                final String[] tokens = str.split(_delimiters);
                for (int i = 1; i < tokens.length; i++) {
                    final String entity = tokens[i];

                    if (facts.containsKey(entity)) {
                        facts.get(entity).add(tokens[0]);
                    } else {
                        facts.put(entity, new
HashSet<>(Arrays.asList(tokens[0])));
                    }
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        }

        return facts;
    }
```

3.4 Словники

Формування словників для роботи з неструктурованими текстами є однією з найскладніших задач, оскільки має враховувати особливості граматики мови, її лексики. Часто у текстах природною мовою можна зустріти «розмовні» слова, які не слід ігнорувати при аналізі, так як вони можуть описувати також стан об’єктів про які йдеться у тексті. У текстах природною мовою часто зустрічаються також скорочення, слова, запозичені з інших мов та неологізми. Також слід враховувати особливості мови, та особливості основної галузі, про яку буде йтися у тестах, що аналізуються, оскільки в таких текстах характерні слова та словосполучення, притаманні лише цій галузі.

Так, наприклад, при розробці словника ключових для туристичної галузі першою проблемою є не велика кількість доробок у цьому напрямку. Тому за

основу словників, що використано у роботі, було взято аналогічні словники іншими мовами, зокрема російською та англійською. Вибір саме таких мов пояснюється їх розповсюдженістю та великою кількістю тематичних словників. Найбільше тематичних словників розроблено англійською мовою, але їх недоліком є те, що граматичні особливості української та англійської мов є досить різними. З мов, що мають схожу на українську мову структуру, найбільше тематичних словників російською мовою. Для перекладу цих словників були задіяні автоматичні методи з відкритих бібліотек, оскільки переклад такої великої кількості слів, яка використана у тематичних словниках «в ручну», є майже неможливим.

Нижче наведені фрагменти коду, що автоматично перекладає текст з однієї мови на іншу.

```
def booking_ids():
    f = open(BOOKING_IN, 'r')
    f_out = open(BOOKING_OUT, 'w')
    for line in f.read().strip().split('\n'):
        line = line.split('\t')
        booking = line[10]
        if len(booking) > 5:
            f_out.write("%s,%s\n" % (booking, line[0]))

def fill_ids():
    global URL_IDS
    f = open(BOOKING_OUT, 'r')
    for line in f.read().strip().split('\n'):
        line = line.split(',')
        URL_IDS[line[0][:-7] + "html"] = line[1]

def parse_xml():
    fill_ids()
    tree = ET.iterparse(XML_IN, events=('end',))
    f_out = codecs.open(XML_OUT, 'w', 'utf-8')
    f_out.write("<txts>")
    h_cnt = 0
    g_cnt = 0
    for event, elem in tree:
        if event == 'end' and elem.tag == 'review':
            h_cnt += 1
            id = None
```

```
            txt = ""
            for chld in elem:
                if chld.tag == 'url':
                    url = chld.text.split('?')[0]
                    if url in URL_IDS:
                        id = URL_IDS[url]
                if chld.tag == 'pros' or chld.tag == 'contras':
                    for c in chld:
                        if c.text is not None:
                            txt += c.text;
                        if id is not None:
                g_cnt += 1
                f_out.write("<rv>")
                f_out.write("<id>" + id + "</id>")
                f_out.write("<txt>" + txt + "</txt>")
                f_out.write("</rv>")
            print "reviews: %d, matches: %d" % (h_cnt, g_cnt)
    f_out.write("</txts>")

def trans(w):
    w = w.replace('\n', '.').split('.')
    txt = None
    for wl in w:
        if len(wl) < 2:
            continue
        try:
            CONN.request("GET", URL + "key=" + KEY + "&text=" + wl + "&lang=en-
ru")
            r = CONN.getresponse().read()
            tr = ET.fromstring(r)

            f = tr.find('text')
            if f is not None:
                txt = f.text if txt is None else txt + "." + f.text
        except Exception as e:
            print "ERROR", e
    return txt

def ya_translate():
    tree = ET.iterparse(TRANSLATE_IN, events=('end',))
    f_out = codecs.open(TRANSLATE_OUT, 'w', 'utf-8')
    f_out.write("<txts>")
    for event, elem in tree:
        if event == 'end' and elem.tag == 'rv':
            id = None
            txt = None
            for chld in elem:
                if chld.tag == 'id':
                    id = chld.text
```

```
            if chld.tag == 'txt':
                to_trans = chld.text
                                txt = trans(to_trans)
        if id is not None and txt is not None:
            f_out.write("<rv>")
            f_out.write("<id>" + id + "</id>")
            f_out.write("<txt>" + txt + "</txt>")
            f_out.write("</rv>")
f_out.write("</txts>")
f_out.close()
```

Розглянемо як приклад фрагменти словників, які були утворені таким методом машинного перекладу зі словників, розроблених іншими мовами. Слова у словниках для туристичної галузі розділено за наступними типами, у дужках після українського перекладу наведено англомовний оригінал:

- місцезнаходження (location);
- рівень обслуговування (service);
- харчування (food);
- еомер (room);
- будівля і територія (hotel);
- Інтернет (internet);
- розваги (entertainment);
- чистота (cleanliness);
- послуги в готелі (services);
- атмосфера (ambiance);
- сімейний готель (familyhotel);
- романтичний готель (romantichotel);
- молодіжний готель (younghotel);

Так об’єкти, які можуть зустрічатися у текстах, що стосуються туристичної сфери в цілому та готелів вчасності, можна розділити за зазначеними вище темами.

Приклад такого розділення наведено нижче:

- автобусна зупинка Місцезнаходження;
- берег моря Місцезнаходження;
- ботанічний сад Місцезнаходження;
- міський пляж Місцезнаходження;
- дрібна галька Місцезнаходження;
- набережна Місцезнаходження;
- біле вино Харчування;
- варені яйця Харчування;
- вино червоне Харчування;
- час обіду Харчування;
- бізнес ланч Харчування;
- обідня перерва Харчування;
- час вечері Харчування;
- двір Будівля і територія;
- головний корпус Будівля і територія;
- бібліотека Будівля і територія;
- будівля Будівля і територія;
- головний ресторан Харчування;
- прасувальна дошка Номер;
- гаряча вода Номер;
- автономне опалення Номер;
- окріп Номер;
- гості готелю Номер;
- двоспальне ліжко Номер;
- проживання Номер;
- індивідуальний трансфер Послуги в готелі;

- контингент відпочиваючих Атмосфера;
- готельний гід Послуги в готелі.

Повна версія обєктів для побудови фраз наведена у додатку В.

Переліки прикметників, що описують ставлення до об’єкту, про який йдеться у тесті, розділено на дві групи: позитивні та негативні. Таке розділення прикметників пояснюється тим, що вони характеризують об’єкт, тобто впливають на загальну оцінку закладу і підвищують її в цілому або понижують.

Так, наприклад, у перелік позитивних прикметників для туристичної галузі входять наведені нижче слова:

- адекватний;
- відповідати вимогам;
- рівноцінний;
- одно цінний;
- рівний;
- відповідний;
- співрозмірний;
- сумірний;
- порівнювати;
- охайний;
- безпечний;
- екологічно чистий.

Повна версія прикметників для побудови фраз наведена у додатку В.

Так, наприклад, у перелік негативних прикметників для туристичної галузі входять наведені нижче слова:

- пекельний;
- сатанинський;
- демонічний;

- диявольський;
- інфернальний;
- страшний;
- біднуватий;
- бідний;
- знедолений;
- небагатий;
- незаможний;
- незабезпечений;
- малозабезпечений;
- несмачний;
- без смаку.

Повна версія прикметників для побудови фраз наведена у додатку В.

Таким чином, з утворених словників можна створити комбінований словник, який містить конструктори фраз, які характерні саме для туристичної галузі, зокрема для опису готелів та обслуговування у них. Приклади таких конструкторів фраз наведені нижче.

- веселий персонал, стюардеса, фрукт, екскурсовод;
- життєрадісний персонал, стюардеса, фрукт, екскурсовод;
- веселий персонал, стюардеса, фрукт, екскурсовод;
- чудова будівлю, номер, коридор, основний ресторан, прибиральник, територія, гаряче, хамам, готель, море, офицант, таксист, ресторан, відпочинок, купання;
- просто чудовий відпочинок, готель, екскурсовод;
- гарний вид, коктель, шлях, екскурсовод;
- працювати тихий кондиціонер;

- коштує копійки вино, морепродукт, пиво, путівка, зв'язок, таксі, фрукт, екскурсія;
- весь день займатися анімаційна команда, анімація;
- хороша зустріч, ціна;
- відмінна зустріч, ціна;
- першокласна зустріч, ціна;
- погана зустріч, ціна;
- безкоштовно возити автобус, адміністрація, анімаційна команда, народ, персонал, пляж, перебування;
- ліцензійний вид, горілка, душова, пиво, програма, турагенція, екскурсовод.

Повна версія конструкторів фраз наведена у додатку XXX.

3.5 База данних

Одним з основних елементів програми, що була розроблена, є база даних до якої звертається програма, так як у цій базі міститься велика частина усіх статичних даних, що використовуються для вилучення та аналізу ключових фактів статей та відгуків про готелі.

База даних містить дані про готелі, відгуки та статті, з якими працюють усі інші модулі системи. Структура основних таблиць та основних полів бази даних наведена на рисунку 8.

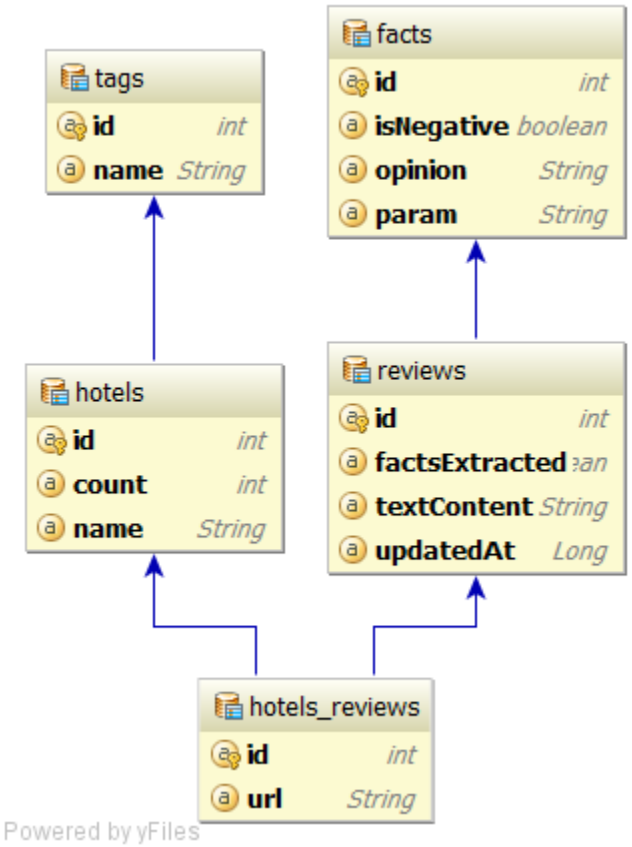


Рисунок 8 - Структура основних таблиць

Основними таблицями бази є такі таблиці:

- hotels_reviews – основна таблиця, яка містить ідентифікатор відгуку або статті та послання на нього. Ця таблиця посилається на таблиці hotels та reviews;
- hotels – таблиця, яка містить ідентифікатор готелю, його назву та кількість звернень до нього. Ця таблиця посилається на таблицю tags;
- tags – таблиця, яка містить факти, зокрема їх ідентифікатор та назву;
- reviews – таблиця, яка містить перелік вилучених фактів, зокрема ідентифікатор множини фактів та текст, з якого вона вилучена. Ця таблиця посилається на таблицю facts;

- facts – таблиця, яка містить факти, показник позитивний чи негативний факт, відсоток яких приходиться на цей факт відносно усіх фактів про цей готель, відсоток готелів, які також мають цей факт, та точність.

3.6 Робота програми

3.6.1 Вилучення основних ключових фактів

Основною задачею розробленої програми є вилучення основних ключових фактів з відгуків та статей про готелі, оцінка вилучених фактів та аналіз їх достовірності.

Розглянемо основні вилучені факти для готеля Amata Resort, що знаходиться у Пхукеті, Таїланд. Вивід має наступі поля.

1. Точність («+», якщо вірно, «-», якщо невірно, «0», якщо не підлягає оцінці);
2. Факт;
3. Негативність факту(«true», якщо факт оцінено як негативний, «false», якщо оцінено факт як позитивний);
4. Відсоток, який складає цей факт від всіх фактів про готель з урахуванням заперечення;
5. Відсоток готелів, які мають у своєму описі цей факт з урахуванням заперечення;

У таблиці 3 наведені основні вилучені факти для готеля Amata Resort.

Таблиця 3 - Основні вилучені факти для готеля Amata Resort

-	інтернет:	платний	false	0.018867924528301886	0.2505003335557038
+	Номер:	просторий	false	0.22641509433962265	0.021014009339559707
0	сніданок:	безкоштовний	false	0.018867924528301886	0.685790527018012

0	готель:	новий	false	0.05660377358490566	0.18845897264843228
-	трансфер:	безкоштовний	false	0.018867924528301886	0.30520346897931955
0	літак:	новий	false	0.018867924528301886	0.07004669779853236
0	готель:	поганий	true	0.03773584905660377	0.36624416277518346
+	Wi-fi:	безкоштовний	false	0.07547169811320754	0.6791194129419613
+	Вулиця:	галасливий	false	0.018867924528301886	0.1447631754503002
0	номер:	чистенький	false	0.018867924528301886	0.2408272181454303
+	Пляж:	знаменитий	false	0.018867924528301886	0.13408939292861907
+	Сейф:	безкоштовний	false	0.018867924528301886	0.3685790527018012
0	пляж:	чистий	false	0.09433962264150944	0.5587058038692462
0	рушник:	новий	false	0.018867924528301886	0.10340226817878585
+	Басейн:	приватний	false	0.018867924528301886	0.0913942628418946
0	номер:	шикарний	false	0.03773584905660377	0.020346897931954638
0	будівля:	новий	false	0.018867924528301886	0.00733822548365577
0	готель:	поганий	false	0.018867924528301886	0.36624416277518346
0	готель:	шикарний	false	0.03773584905660377	0.4306204136090727
0	меблі:	старий	false	0.05660377358490566	0.39859906604402934
0	номер:	розкішний	false	0.018867924528301886	0.18178785857238158
0	пляж:	прекрасний	false	0.03773584905660377	0.39693128752501666
0	море:	шикарний	false	0.018867924528301886	0.19879919946631086
0	екскурсія:	безкоштовний	false	0.018867924528301886	0.14076050700466977
0	човен:	швидкісний	false	0.018867924528301886	0.0513675783855904
0	відпочинок:	шикарний	false	0.018867924528301886	0.1180787191460974
0	готель:	розкішний	false	0.018867924528301886	0.43862575050033353

Повна версія таблиці наведена у додатку В.

Для порівняння наведемо опис даного готелю з ресурсу

«https://www.booking.com/». Короткий опис готелю та опис готелю у якості скрішота наведено нижче на рисунку 9.

Опис готелю.

Курортний готель EEd Amata розташований у місті Банг-Саре, за 9 км від аквапарку Cartoon Network Amazone. До послуг гостей номери з відкритим басейном, безкоштовна приватна автостоянка, бар і спільний лаунж. До послуг гостей обслуговування номерів, а також дитячий ігровий майданчик. До послуг гостей цілодобова стійка реєстрації, трансфер до / з аеропорту, спільна кухня та безкоштовний Wi-Fi.

Гості курортного готелю можуть замовити страви з меню або американський сніданок.

Курортний готель EEd Amata пропонує приладдя для барбекю. Околиці популярні для велосипедних прогулянок. У помешканні можна взяти напрокат автомобіль.

Аквапарк RamaYana розміщений за 18 км від курортного готелю EEd Amata. Найближчий міжнародний аеропорт У-Тапао Районг-Паттайя розміщений за 17 км від курорту.

Як видно з наведеного вище списку, для цього готелю неможливо перевірити достовірність більшості отриманих фактів. Це пов’язано з тим, що у офіційних даних готеля дані словосполучення не зустрічаються, також причиною цього є те, що багато з фактів мають суб’єктивну складову, так, наприклад, факт «готель: шикарний», «відпочинок: шикарний» або «готель: розкішний». Також частина фактів не стосується на пряму відгуку про готель, а скоріше лише відгуку про весь відпочинок в цілому, так, наприклад, «літак: новий», «човен: швидкий» або «екскурсія: безкоштовна». Також слід зауважити, що більшість вилучених фактів є позитивно охарактеризованими у рамках цього аналізу. Частота появи більшості фактів у всій множині статей та відгуків, що аналізується, є досить

високою, що свідчить про те, що багато користувачів відмічають ці особливості готелю.

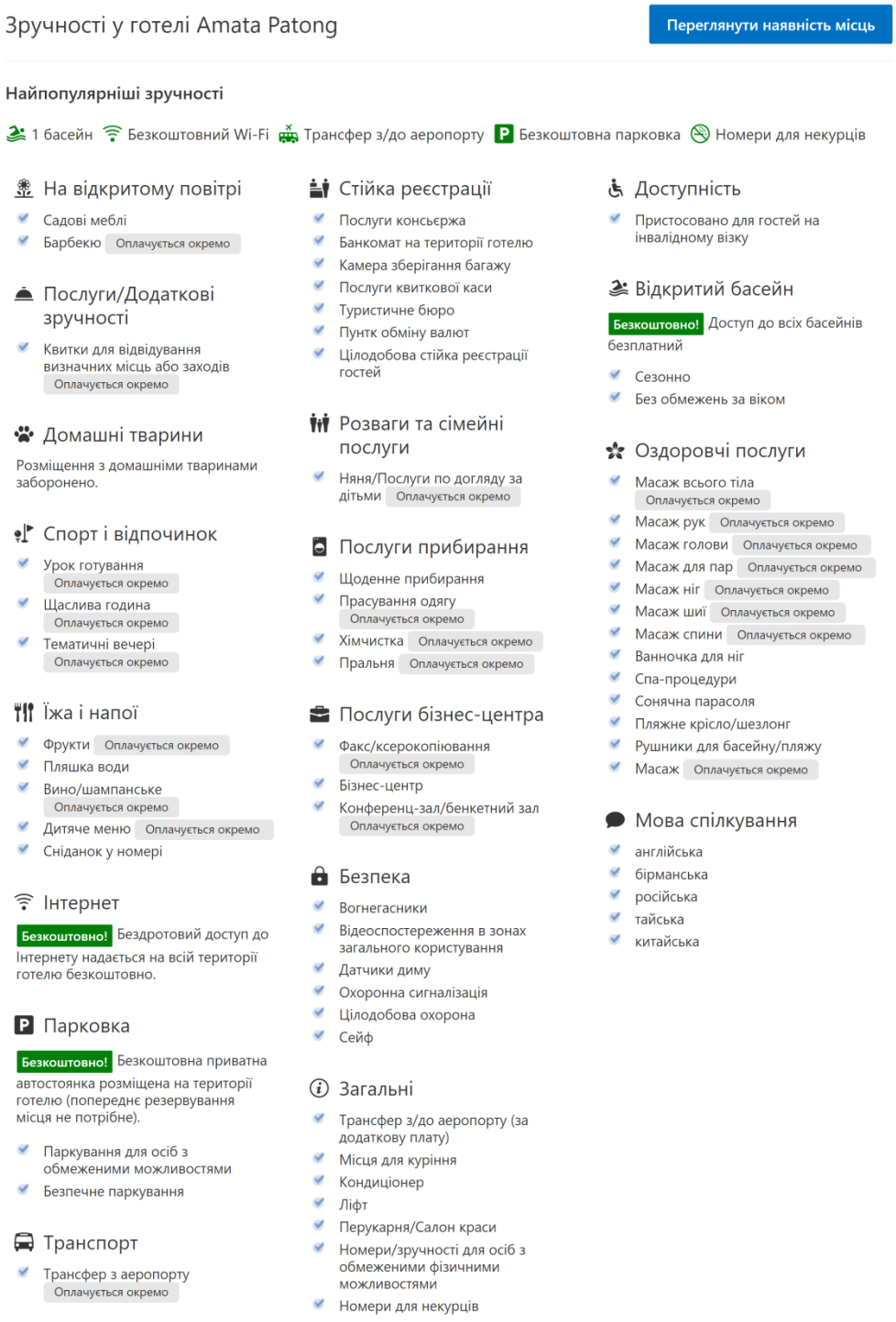


Рисунок 9 - Опис готелю

Як видно з наведеного вище списку, для цього готелю неможливо перевірити достовірність більшості отриманих фактів. Це пов’язано з тим, що у офіційних даних готеля дані словосполучення не зустрічаються, також причиною цього є те, що багато з фактів мають суб’єктивну складову, так, наприклад, факт «готель: шикарний», «відпочинок: шикарний» або «готель: розкішний». Також частина фактів не стосується на пряму відгуку про готель, а скоріше лише відгуку про весь відпочинок в цілому, так, наприклад, «літак: новий», «човен: швидкий» або «екскурсія: безкоштовна». Також слід зауважити, що більшість вилучених фактів є позитивно охарактеризованими у рамках цього аналізу. Частота появи більшості фактів у всій множині статей та відгуків, що аналізується, є досить високою, що свідчить про те, що багато користувачів відмічають ці особливості готелю.

Розглянемо основні вилучені факти ще для одного готелю. Нижче наведена таблиця 4 з ключовими фактами для готелю «Camping Village Cavallino», що знаходитися у Венеції, Італія. Поля мають ті ж самі характеристики, що й в таблиці 3, яка наведена вище.

Таблиця 4 - Основні вилучені факти для готелю «Camping Village Cavallino»

+	Інтернет:	платний	false	0.030303030303030304	0.2505003335557038
-	сніданок:	безкоштовний	false	0.030303030303030304	0.685790527018012
0	дорога:	прекрасний	false	0.030303030303030304	0.027685123415610406
0	трансфер:	безкоштовний	false	0.030303030303030304	0.30520346897931955
+	Пляж:	пісочний	false	0.06060606060606061	0.22915276851234156
0	Готель:	поганий	false	0.030303030303030304	0.36624416277518346
+	Лежак:	безкоштовний	false	0.030303030303030304	0.25883922615076715
0	Бар:	безкоштовний	false	0.030303030303030304	0.1057371581054036
0	Місто:	новий	false	0.030303030303030304	0.10507004669779853

+	Пляж:	приватний	false	0.12121212121212122	0.3845897264843229
-	wi-fi:	безкоштовний	false	0.12121212121212122	0.6791194129419613
+	Територія:	просторий	false	0.030303030303030304	0.12975316877918613
-	Інтернет:	безкоштовний	false	0.06060606060606061	0.5443629086057371
0	Природа:	незайманий	false	0.030303030303030304	0.07071380920613743
0	Пляж:	шикарний	false	0.09090909090909091	0.26717811874583053
+	Інтернет:	бездротовий	false	0.030303030303030304	0.16244162775183454
+	Пляж:	безкоштовний	false	0.030303030303030304	0.20413609072715144
0	готель:	розкішний	false	0.030303030303030304	0.43862575050033353
+	Пляж:	чистий	false	0.15151515151515152	0.5587058038692462

Повна версія таблиці наведена у додатку XXX.

Для порівняння наведемо опис даного готелю з ресурсу «https://www.booking.com/». Опис готелю у якості скрішота наведено нижче на рисунку 10.

Опис готелю:

До послуг гостей цього кемпінгу, розташованого на півострові Кавалліно, в оточенні сосен і оливкових дерев, власний пляж, басейн, гідромасажна ванна на відкритому повітрі, а також поле для міні-гольфу та комфортабельні пересувні будинки.

Гостям пропонується розміщення в сучасних, чудово укомплектованих пересувних будинках з патіо і холодильником. Всі номери обладнані власними ванними кімнатами, а деякі номери оснащені міні-кухнею і обідньою зоною.

У кемпінгу Village Cavallino проводяться різноманітні розважальні програми для гостей будь-якого віку. Спеціально для самих маленьких гостей на території кемпінгу облаштовано дитячий ігровий майданчик та ігрова кімната. Також в кемпінгу працює прокат велосипедів, мінімаркет і супермаркет.

Крім цього, в розпорядженні гостей кемпінгу Cavallino Village безкоштовна парковка і зона для мийки собак (з самообслуговуванням). У кемпінгу, при

будівництві якого використовувалися екологічно чисті матеріали, використовується переважно сонячна енергія.

Кемпінг розташований на півострові, який відокремлює Венеціанську лагуну від моря, в 6 км від човнового терміналу Пунта-Саббіону, звідки можна доїхати до венеціанських островів Лідо. Поїздка від кемпінгу до міста Лідо-ді-Єзоло займає 25 хвилин.

З отриманих даних також можна зробити наступні висновки. Факти, які не піддаються перевірці, також не вказані у описі даного готелю. Переважно, користувачі у відгуках звертають увагу на позитивні факти. Також слід зазначити, що більшість фактів, які було вилучено зі статей та відгуків, є достовірними для даного готелю. Найчастіше серед вилучених фактів зустрічаються такі факти «Місто:новий», «wi-fi: безкоштовний» та «Пляж: чистий».

Перевагою розробленого в даній роботі алгоритму є можливість вилучити основні ключові факти з множини неструктурованих текстів, оцінити їх достовірність. Як видно з наведених вище таблиць, частоти, з якими ці факти зустрічаються у статтях та відгуках кожного готелю та частоти, з якими вони зустрічаються у множині всіх готелів, є досить високими. Це свідчить про коректність роботи алгоритму та те, що переважно люди у своїх статтях та відгуках звертають увагу на одні й ті ж самі факти.

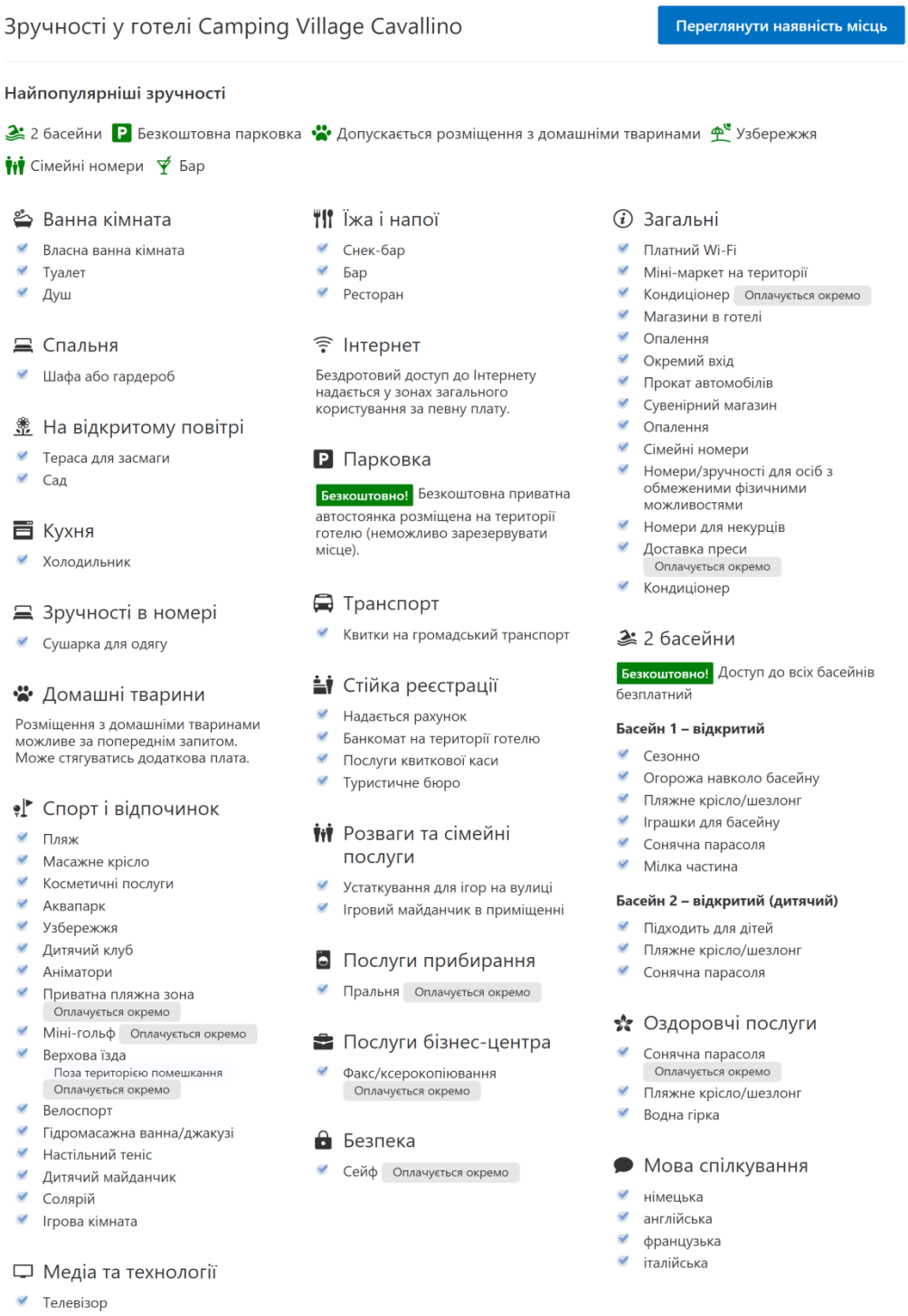


Рисунок 10 - Опис готелю

3.6.2 Вилучення специфічних ключових фактів

Для вилучення специфічних ключових фактів використано алгоритм TFIDF. Особливістю цього алгоритму є те, що з його допомогою можна виділити факти, що притаманні лише для даного об'єкту і не притаманні для інших подібних. Тобто, у прив'язці до туристичної галузі і готелів зокрема, цей алгоритм виділяє особливості та специфікацію цього конкретного готелю.

Розглянемо на прикладі готелю Abalone Resort, що знаходиться на Гоа, Індія. Вилучені зі статей та відгуків про цей готель за допомогою алгоритму TFIDF специфічні ключові факти наведені нижче у таблиці. Таблиця має наступні колонки: об'єкт, його опис, характеристика даного факту та відсоток випадків у яких у статтях було вилучено цей факт. Розглянемо вилучені факти у таблиці 5, що наведена нижче.

Таблиця 5 – Ключові факти для готелю Abalone Resort

персонал:	надійний	false	0.008201569432013972
переживання:	розсіятися	false	0.008201569432013972
індію:	материковий	false	0.007921730337089395
індію:	звичайний	false	0.007921730337089395
компанія:	передвиборний	false	0.007921730337089395
російська:	невгамовний	false	0.007921730337089395
гоа:	північний	false	0.0077231992580000285
візит:	діловий	false	0.007379842785763879
штат:	маленький	false	0.007195476583741021
ходьба:	тихий	false	0.006337384269671515
мініхолодильник:	стельовий	false	0.006337384269671515
відвідування:	дворазовий	false	0.006337384269671515
телик:	буде	false	0.006337384269671515

пропозиція:	заінтригувати	false	0.006337384269671515
випивка:	дешевий	false	0.005864099680370776
можливість:	шукати	false	0.005788682761213217
країна:	зачепити	false	0.005788682761213217
яєчня:	смачний	false	0.005467712954675981
відпочинок:	вийти	false	0.005242084383099478
колонія:	португальська	false	0.005008175044029391
москіт:	жерти	false	0.004753038202253637
ситуація:	зайвий	false	0.004753038202253637
Літера:	значить	false	0.004753038202253637
урок:	англійська	false	0.004753038202253637
джем:	нарізаний	false	0.004753038202253637
середина:	дешево	false	0.004753038202253637
збентеження:	єдиний	false	0.004753038202253637
рідина:	містити	false	0.004753038202253637
екскурсовод:	полоти	false	0.004753038202253637

Повна версія таблиці наведена у додатку В.

Для порівняння наведемо опис даного готелю з ресурсу «https://tophotels.ua/».

Abalone Resort - недорогий готель, побудований на популярному пляжі Baga Beach. Готель підійде як для сімейного відпочинку, так і для ділового візиту, завдяки зручному розташуванню щодо основних визначних пам'яток і транспортних вузлів цього регіону.

Місцезнаходження:

Готель розташований в місті Arpora, в 15 км від ж / д станції. Міжнародний аеропорт знаходиться в 45 км від готелю.

Кількість номерів:

76 номерів.

Типи номерів:

Стандартні номери.

Опис номерів:

Всі номери відповідають міжнародним стандартам, оформлені просто, але чисто й акуратно, укомплектовані зручними меблями.

- балкон;
- ТБ з супутниковими каналами;
- холодильник;
- кондиціонер;
- телефон;
- ванна кімната з душовою кабіною і туалетним приладдям.

Інфраструктура готелю:

- конференц зал.

Типи харчування:

- сніданок.

Безкоштовний сервіс:

- цілодобове обслуговування;
- парасольки і шезлонги біля басейну і на пляжі.

Платний сервіс:

- сейф у адміністратора;
- обслуговування номерів;
- пункт обміну валют;
- пральня;
- організація екскурсій;
- виклик лікаря.

Розваги і спорт:

- фітнес центр;

- відкритий басейн;

- джакузі.

Для дітей:

- дитячий басейн.

Ресторани, бари:

- два сучасних ресторану, які познайомлять з вишуканими стравами різних країн світу.

пляж:

Піщаний, громадський пляж.

Розглянемо отримані результати.

Для порівняння наведемо опис даного готелю з ресурсу «https://www.booking.com/». На рисунку 11 зображено скріншот з сайту з переліком особливостей готелю.

Проаналізувавши вилучені факти та порівнявши їх з інформацією, яка наведена на туристичних ресурсах мережі Інтернет, можна дійти до висновків, що алгоритм вилучення специфічних ключових слів TFIDF виділив зі статей та відгуків факти, характерні лише для цього готелю. Частота, з якою зустрічається дані факти у множині відгуків та статей, що були проаналізовані, наведена у останній колонці таблиці. Значення у цій колонці є досить маленькими, що й показує коректність роботи алгоритму.

Перевагою цього алгоритму є те, що з його допомогою можна вилучити факти, що характерні саме для цього готелю, що може бути корисним для пошуку та підбору готелю за якимись особливими умовами.

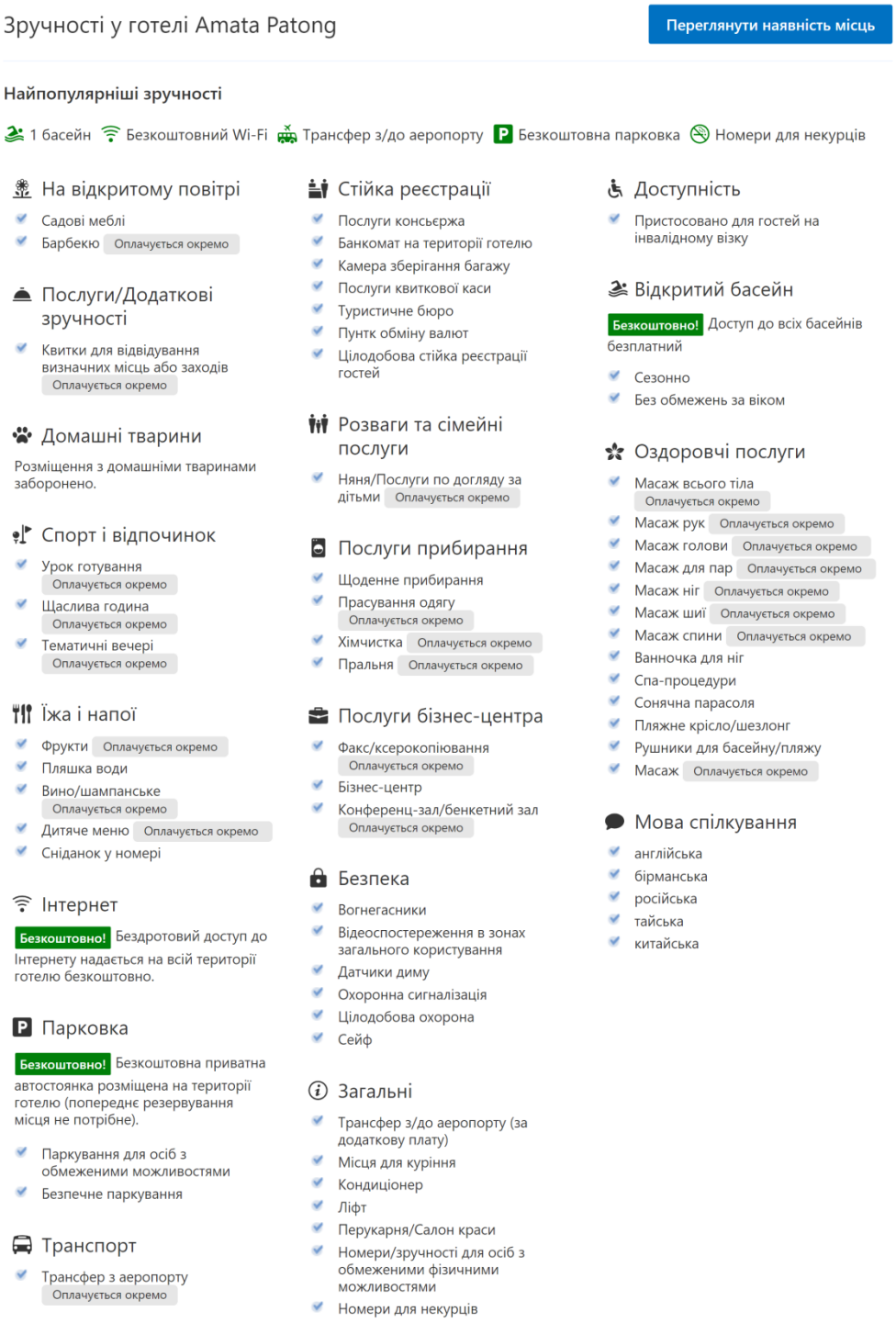


Рисунок 11 - Опис готелю

Недоліком даного методу є те, що важко перевірити достовірність вилучених фактів, оскільки більшість з них не вказана в описі готелю на його офіційних сторінках на туристичних ресурсах. Також у вилучених фактах

залишилось досить багато фактів, що вказують на суб’єктивну думку оповідача, такі факти недоцільно використовувати у рекомендаційних системах. Деяка частина фактів не стосується безпосередньо опису готелю, а описує відпочинок загалом, що пояснюється використанням для аналізу реальних відгуків, у яких не встановлено обмежень на теми, про які дозволено писати, та використанням словників для туристичної галузі в цілому, а не лише для готелів.

3.6 Порівняння алгоритмів

У роботі було проаналізовано відгуки про готелі за допомогою двох алгоритмів, розробленого у ході виконання роботи алгоритму та алгоритму TFIDF. Було використано два алгоритми, щоб мати змогу порівняти отримані результати щодо вилучення ключових фраз з статей та відгуків.

З вже існуючих алгоритмів для порівняння було обрано саме алгоритм TFIDF через те, що він використовує принцип залучення словників, так само як і розроблений алгоритм, та потенційно є корисним для побудови саме рекомендаційних списків через те, що виділяє з множини всіх існуючих фактів лише факти, притаманні для даного готелю, тобто факти, що вирізняють його з поміж інших.

Особливістю розробленого в дані роботі алгоритму є використання тематичних словників, що дає перевагу над деякими вже існуючими. За рахунок використання тематичних словників розроблений алгоритм та алгоритм TFIDF виділяють ключові фрази які стосуються лише заданої тематики, тобто туристичної галузі. Перевагою такого принципу є те, що множина вилучених фактів містить як найменше «зайвих» фактів та не потребує подальшої фільтрації за тематикою. Недоліком такого принципу є недосконалість словників, зокрема те, що багато фраз, утворених за допомогою таких словників несуть у собі суб’єктивну думку оповідача, а отже їх неможливо використовувати для

побудови рекомендаційних списків для Інтернет-ресурсів. Також недоліком існуючих словників є те, що вони не вузько направлені, так, наприклад, словники, що використані у даній роботі використовують словники для туристичної галузі в цілому, а не для опису готелів та готельних послуг. Через цей нюанс у вилучених фактах фігурують факти про літак, стюардів, послуги туристичних агентств тощо.

Однією з потенційних переваг алгоритму TFIDF є те, що він виділяє факти, притаманні одному готелю, але не притаманні усім іншим. Потенційно пошук за такими фактами міг би бути простим, але так, як отримані факти є досить специфічними і не загальновживаними, то такий перелік складно застосувати для пошукових або фільтраційних систем ресурсів, які мітять переліки та характеристики готелів.

Недоліком алгоритму TFIDF є також те, що він не має явних обмежень щодо кількості вилучених фактів. З одного боку ця особливість дає досить великий список фактів, які б могли описувати особливості готелю, але з іншого боку, ця кількість фактів занадто велика, щоб користувач зміг її усю переглянути та обрати необхідні для себе параметри. Також слід зауважити, що більшість, отриманих цим методом фактів, є досить не популярними у всіх відгуках, як правило, це деякі помилки, які користувач випадково допустив, чи жаргонізми, тобто використовувати ці факти недоречно для рекомендаційних систем.

Перевагою розробленого алгоритму є те, що більшість отриманих за його допомогою фраз для кожного конкретного готелю є досить популярними в цілому у всій множині відгуків та статей. Так, наприклад, легко помітити що серед вилучених фактів часто зустрічаються факти, які стосуються якості та вартості послуги доступу у мережу Інтернет, так, наприклад, факти «Інтернет платний», «Інтернет швидкий» або «wi-fi безкоштовний» тощо. Також досить популярними фактами є факти про якість та вартість харчування, привітливість

та охайність персоналу тощо. Як результат, такі списки фактів легко використовувати для рекомендаційних та фільтраційних систем завдяки тому, що через повтори фактів для різних готелів загальна кількість отриманих фактів не є неосяжною, як у алгоритму TFIDF, і може бути переглянута кожним користувачем.

Також за рахунок встановлення бар'єру для якості вилучених фактів зменшується кількість вилучених фактів для кожного готелю. Це потенційно може зменшити усю множини фактів, за якою може відбуватися фільтрація або побудова рекомендацій, що спрощує подальшу роботу з цими фактами та їх обробку.

За рахунок вилучення фактів, які є популярними у всій множині обробленого тексту, зони перетину множини вилучених для кожного конкретного факту є досить великими, один якийсь факт може бути спільним для досить великої кількості готелів. Це може спростити пошук користувачем по системі рекомендацій, побудованій за списками цих вилучених фактів.

Особливістю розробленого в роботі алгоритму є оцінка достовірності вилученого факту на основі офіційних даних про готель. Це має декілька ефектів на потенційне застосування списків вилучених фактів. З одного боку факти, які не підтверджені офіційною інформацією, можуть вважатися модераторами ресурсу не достовірними і просто не використовуватись при побудові рекомендацій для користувачів ресурсу. З іншого боку, відгуки та статті можуть відображати реальний стан готелю, а не той, що вказаний в офіційному описі, тобто можуть бути більш корисними для користувачів ресурсу.

Факти, достовірність яких не вдалося перевірити за рахунок офіційних даних про готель, є, як правило, досить цікавими та корисними для користувачів, оскільки розповідають більше інформації про готель або допоюють офіційну. Так як більшість цих фактів побудована на відгуках користувачів, то вони

висвітлюють відповіді саме на ті питання, які цікавлять відвідувачів туристичних ресурсів.

Завдяки побудові фактів, як словосполучень, що складаються з об’єкту та його характеристики, легко утворити фільтри чи рекомендації на основі цих фактів.

Проаналізувавши переваги та недоліки двох алгоритмів можна прийти до висновку, що розроблений алгоритм більш пристосований для використання у формуванні списків рекомендацій та фільтрації на ресурсах з великою кількістю тестів.

3.7 Перспективи розвитку

Кожен алгоритм має перспективні напрямки розвитку та впровадження у життя користувачів або інші розробки. Для визначання елементів алгоритму, які доцільно було б покращити, слід розглянути найслабші його місця.

Потенційно, одним з найслабших місць є використання тематичних словників через те, що їх потрібно постійно оновлювати, доповнюючи новими словами, які описують нові винаходи, та речі, що входять у вжиток, та виключаючи застарілі слова. Створення більш спеціалізованих словників також може покращити роботу розробленого алгоритму так, як виключить зі списків вилучених фраз фрази, що не стосуються безпосередньо основної теми тестів, зо аналізуються. Зменшення емоційно забарвлених слів у словниках теж може підвищити якість вилучених фактів за рахунок зменшення впливу суб’єктивної думки користувача на множину вилучених фактів.

Для покращення роботи алгоритму було б не погано розробити модуль для корекції помилок, які користувачі допускають при написанні тексту, оскільки велика кількість помилок може спотворювати картину вилучених ключових фактів.

Перспективним також є напрямок створення співвідношень між фактами та об’єднання їх за змістом. Так, наприклад, факти «Інтернет безкоштовний» та «wi-fi безкоштовний» хоча і не є рівносильними з технічної точки зору, але для відвідувачів готелю ці факти означають одне й те саме – безкоштовний доступ до мережі Інтернет. Було б добре об’єднати ці факти тому, що через їх різноманітність вони можуть не потрапити у рейтинг найпопулярніших фактів вилучених з тексту окремо один від одного.

Цікавим напрямком для розвитку є також додання модулю, що зміг би динамічно оновлювати вилучені факти при створенні нових текстів. Таким чином, при додаванні нових статей та відгуків про готелі системи могли б миттєво на це реагувати та оновлювати оновлювати рекомендації для користувачів.

Перспективним напрямком розвитку є додавання впливу дати створення тексту на вагу значущості вилучених з нього фраз. Необхідність цього пояснюється тим, що з плином часу характеристики об’єктів можуть змінюватись і старий опис цих об’єктів може бути вже не актуальним. За рахунок великою кількості більш старих текстів з хибною інформацією, хибні данні можуть потрапляти у списки вилучених фактів та спотворювати актуальну картину.

Також одним з можливих напрямків розвитку розробленого алгоритму є збільшення кіот кості варіантів достовірності вилученого факту. Так, наприклад, можна було б факт «пляж кам’яний», вилучений з фактів, прирівняти до факту «пляж з гальки» для факту з офіційного опису готелю, так я ці факти є дуже близькими за своїми смисловими значеннями для відвідувачів.

Одним з напрямків розвитку створеного алгоритму є збільшення його швидкодії. Це зробить його легшим для впровадження у реальних системах, оскільки його використання займе менше часу та обчислювальної потужності.

Дані, отримані за допомогою розробленого алгоритму, можна

використовувати не лише для формування фільтрів на ресурсах з великою кількістю статей та побудови списків рекомендацій для користувачів таких ресурсів. Також, списки фактів, вилучені за допомогою розробленого алгоритму, можна використовувати для побудови статистики та аналізу того, що цікавить користувачів, які пишуть відгуки, або авторів статей. Так на основі вилучених кочових фактів зі статей та відгуків про готелі можна зрозуміти, на які особливості готелю та відпочинку в цілому користувачі Інтернет-ресурсів звертають найбільше уваги, та які особливості готелів є найбільш пріоритетними для відвідувачів. Найбільше покрашень у роботу алгоритму можна внести покращивши тематичні словники, які він використовує у своїй роботі. А найперспективнішими напрям рямками для впровадження є системи рекомендацій ресурсів які містять велику кількість неструктурованих текстів, наприклад статей та відгуків, або аналітична робота для прогнозування очікувань та потреб користувачів таких ресурсів.

ВИСНОВКИ РОЗДІЛУ 3

Третій розділ роботи присвячений безпосередньо реалізації розробленого алгоритму. У цьому розділі описується структура проекту, наведений перелік основних елементів створеної програми та описано їх використання. Також у цьому розділі описується створення тематичних словників, що використовуються для вилучення основних ключових фактів із тексту. У роботі описаний як і самий процес формування словників, так і складності, які виникають у процесі створення тематичних словників українською мовою.

У роботі реалізовано два алгоритми вилучення ключових слів: розроблений у ході виконання роботи алгоритм та алгоритм TFIDF. Наведено приклади списків вилучених фраз за допомогою кожного з алгоритмів та проведено порівняння списків. Було виявлено, що обидва алгоритми правильно виділяють ключові фрази з неструктурованих текстів. Також була наведена частота, з якою вилучені факти зустрічаються у всій множині текстів, що аналізуються.

Одним з елементів розділу є порівняння двох використаних алгоритмів, а саме їх переваг та недоліків. Для формування списків рекомендацій на ресурсах, зо спеціалізуються на статтях про готелі більше підходить розроблений у ході виконання роботи алгоритм так як він виділяє факти лише заданої тематики, у даному випадку туристичної тематики, вилучає невелику кількість основних ключових фактів для кожного готелю та оцінює достовірність цих фактів.

Також у даному розділі наведено перспективні напрямки розвитку щодо поліпшення створеного алгоритму та потенційні сфери його впровадження.

ВИСНОВКИ

На сьогоднішній день існують інструменти для виділення з текстів природною мовою різних синтаксично коректних словосполучень, відповідних граматиці цієї мови. На основі існуючих способів вилучення фраз з неструктурованих текстів був розроблений алгоритм вилучення основних ключових фраз з неструктурованих текстів який має більшу швидкодію та ефективність за існуючі аналоги.

Розроблений спосіб дає можливість швидко та ефективно отримувати ключові україномовні факти з неструктурованих текстів природною мовою. Також особливістю розробленого способу є те, що він виявляє ключові факти лише за тематикою тексту, що значно спрощує подальшу роботу з цими ключовими фактами.

В ході дипломної роботи було вироблено докладне дослідження методів з автоматичного вилучення ключових фраз з наукових статей англійською мовою.

Був запропонований і розроблений власний метод автоматичного вилучення ключових фраз. Запропонований метод полягав у використанні словника в якості використано перекладені на українську мову тематичні словники. Також був реалізований стандартний метод автоматичного вилучення ключових фраз.

Методи, що були розроблені та досліджені у даній роботі призначені для вилучення ключових фраз з статей та відгуків про готелі, але за рахунок своєї універсальності вони можуть бути застосованим й в інших галузях У роботі реалізовано два алгоритми вилучення ключових слів: розроблений у ході виконання роботи алгоритм та алгоритм TFIDF. Наведено приклади списків вилучених фраз за допомогою кожного з алгоритмів та проведено порівняння списків. Було виявлено, що обидва алгоритми правильно виділяють ключові фрази з неструктурованих текстів. Також була наведена частота, з якою вилучені

факти зустрічаються у всій множині текстів, що аналізуються.

У практичній частині роботи реалізовано два алгоритми вилучення ключових слів: розроблений у ході виконання роботи алгоритм та алгоритм TFIDF. За наведеними прикладами списків вилучених фраз за допомогою кожного з алгоритмів було проведено порівняння якості вилучення фраз та оцінка можливості використання розробленого алгоритму для побудови списків рекомендацій.

В якості подальших напрямків розвитку роботи можна виділити наведені нижче напрямки:

- використання машинного перекладу в більш широких масштабах;
- перенесення досліджень на тексти іншими мовами;
- оптимізація алгоритмів для роботи з більш об'ємними наборами даних;
- поліпшення точності вилучення фактів;
- розробка більш вузькоспеціалізованих тематичних словників.

Список використаної літератури

1) Світовий банк. Користувачі Інтернету, 2018. URL: http://data.worldbank.org/indicator/IT.NET.USER.P2 (дата звернення 05.08.2020).
2) Крістофер Д. Меннінг, Прабхакар Рагаван, Гінріх Шютце. Вступ до пошуку інформації. - Cambridge University Press, 2008. .
3) Крістофер Д. Меннінг, Прабхакар Рагаван, Гінріх Шютце. Вступ до пошуку інформації. - Cambridge University Press, 2008.
4) Stanford CoreNLP: Набір основних інструментів NLP. - 2015. - URL: http://nlp.stanford.edu/software/corenlp.shtml (дата звернення 05.08.2020).
5) Беренд Г., Фаркас Р. ESZTERGOM: Розробка функцій для вилучення ключових фраз. - Асоціація обчислювальної лінгвістики. 2010
6) Лопес П., Ромарі Л. ХУМБ: Автоматичне вилучення ключових термінів з наукових статей у GROBID.
7) Нгуєн Т.Д., Луонг М-Т. WINGNUS: Вилучення ключової фрази з використанням логічної структури документа . - Асоціація обчислювальної лінгвістики. 2010.
8) Кляйн Д., Крістофер Д. Меннінг Точний нелексикалізований синтаксичний розбір . - Асоціація обчислювальної лінгвістики. 2003.
9) Witten I.H., Paynter G.W., Frank E., Gutwin C., Craig G. Nevill-Manning Kea: Практичне автоматичне вилучення ключових фраз . - ACM DL. 1999.
10) Лопес П. GRISP: поєднання автоматичного розпізнавання бібліографічних даних та вилучення термінів для наукових публікацій . - ECDL. 2009.

11) Парв В, Бурд Р.Д., Богураєв В.К. Автоматичне вилучення глосарію: поза термінологічною ідентифікацією . - Асоціація обчислювальної лінгвістики. 2002.

12) Лопес П., Ромарі Л. GRISP: масивна багатомовна термінологічна база даних для науково-технічних областей . - Європейська асоціація мовних ресурсів. 2010.

13) Ченг С-С., Лін С.-Д. LIBSVM: бібліотека для підтримки векторних машин. Технічний звіт. 2001 рік.

14) Врайтен І.Н., Франк Е. Data Mining: Практичні засоби та методи машинного навчання . - Elsevier Inc, 2005.

15) Мао С., Розенфельд А., Канунго Т. Алгоритми аналізу структури документів: опитування літератури. - SPIE Electronic Imaging Conference. 2003.

16) Luong M-T., Nguyen T.D., Kan M-Y. Відновлення логічної структури в наукових статтях з багатими характеристиками документів . - Міжнародний журнал цифрових бібліотечних систем (IJDLS). 2010.

17) Холл М., Френк Е., Холмс Г., Пфарінгер Б., Ройтеманн П., Віттен І.Х. Програмне забезпечення для обробки даних WEKA: оновлення . - Інформаційний бюлетень про дослідження ACM SIGKDD. 2009. Том 11. Випуск 1.

18) Сегаран Т. Програмування колективного інтелекту . - O'Reilly Media. 2007.